Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 1391801, 14 pages
https://doi.org/10.1155/2021/1391801

Research Article

Joint Generative Image Deblurring Aided by Edge Attention Prior and Dynamic Kernel Selection

Zhichao Zhang,[1] **Hui Chen**,[2] **Xiaoqing Yin**,[3] **and Jinsheng Deng**[3]

[1]*College of Computer, National University of Defense Technology, Changsha 410000, China*
[2]*Science and Technology on Integrated Logistics Support Laboratory, National University of Defense Technology, Changsha 410000, China*
[3]*College of Advanced Interdisciplinary Studies, National University of Defense Technology, Changsha 410000, China*

Correspondence should be addressed to Zhichao Zhang; 1933978660@qq.com

Received 27 May 2021; Accepted 2 July 2021; Published 1 August 2021

Academic Editor: Jerry Chun-Wei Lin

Image deblurring is a classic and important problem in industrial fields, such as aviation photo restoration, object recognition in robotics, and autonomous vehicles. Blurry images in real-world scenarios consist of mixed blurring types, such as a natural motion blurring owing to shaking of the camera. Fast deblurring does not deblur the entire image because it is not the best option. Considering the computational costs, it is also better to have an alternative kernel to deblur different objects at a high-semantic level. To achieve better image restoration quality, it is also beneficial to combine the blurring category location and important structural information in terms of specific artifacts and degree of blurring. The goal of blind image deblurring is to restore sharpness from the unknown blurring kernel of an image. Recent deblurring methods tend to reconstruct prior knowledge, neglecting the influence of blur estimation and visual fidelity on image details and structure. Generative adversarial networks(GANs) have recently been attracting considerable attention from both academia and industry because GAN can perfectly generate new data with the same statistics as the training set. Therefore, this study proposes a generative neural architecture and an edge attention algorithm developed to restore vivid multimedia patches. Joint edge generation and image restoration techniques are designed to solve the low-level multimedia retrieval. This multipath refinement fusion network (MRFNet) can not only perform deblurring of images directly but also individual the frames separately from videos. Ablation experiments validate that our generative adversarial network MRFNet performs better in joint training than in multimodel. Compared to other GAN methods, our two-phase method exhibited state-of-the-art performance in terms of speed and accuracy as well as has a significant visual improvement.

1. Introduction

GANs have exhibited a promising performance on edge restoration and image deblurring [1, 2]tasks. However, restoration methods typically introduces artifacts if the blurred area has uniform intensity, because it selects an incorrect region for deblurring. Deep learning approaches have been proposed to handle complex natural blurring. These methods use convolutional layers to extract features by scanning blurred and sharp images and subsequently fusing features with deconvolution layers and recording the learned results [3–5]. Xu et al. [6], Schuler et al. [7], and Zhang et al. [8] adopted this two-stage traditional procedure based on the use of an encoder-decoder neural network. However, these methods still adopt the traditional framework with low prediction performance.

Inspired by the problems described above, Kupyn et al. [9] designed a new framework for deblurring that could calculate the differences between generative and original images. GANs have shown promising performance in image deblurring. Scholars have also achieved significant improvements using other complicated GAN networks, such as DeblurGAN [9], DeblurGANv2 [10], and EGAN [1, 2, 11, 12]. However, a GAN requires a large amount of computational and memory resources when comparing the generated and original images of the discriminator. With advancements

in the design of complicated network models, more complex end-to-end deep learning approaches have been proposed for deblurring. These networks can be divided into four classes: including multiscale, recurrent, multipatch, and scale-iterative networks.

The frameworks of Nah et al. [13] and Lin et al. [14] employ a multiscale style. The main idea of their frameworks is the implementation of the coarse-to-fine strategy to deblur images in consecutive stages. The coarse stage obtains features by using scales, and the features are then 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 is performed directly via the scale-cascaded structure. However, despite the achievement of suitable results, such networks size and depth eventually become excessive, thus leading to increased graphics processing unit (GPU) memory consumption.

Tao et al. proposed the recurrent architecture in which subsequentthe next rounds of training can be aided by the results of the previous round [15]. Multipatch networks have been proposed by Nekrasov et al. [16] and Zhang et al. [17], whereby the recurrent method was applied by regarding the last-turn results as the next round input for refining 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 be conducive to the reduction of the parameters by learning from patches in a single round. However, the method is not stable for a complex blurring.

Ye et al. [18] used the scale-iterative architecture to train the model by applying an upsampling path with the aid of the results of the previous iteration, blurring kernels varied in different regions. Low-frequency information was present, such as semantic and category contents, and background color, along with high-frequency information, such as edge and structure. High-spatial gradients are diminished more in blurred or low-resolution images. Hence, we combine the ideas of multiscale and recurrent architectures to produce a new framework. The design of the MRF network overcomes the parameter and low-efficiency issues of multiscale and recurrent architectures, respectively. Considering the above limitations and strengths, we propose a multipath refinement network called MRFNet. The main contributions of this study are summarized as follows.

Firstly, in terms of the network, we develop a multipath refinement network (MRFNet) for joint low-level image training, with a plug-and-play feature for multiple attention modules. It is plug and play for several edge detection networks for image information prior and feature extraction, and multiple attention modules can also be added at multiscale dataflow paths. An iterative and recurrent strategy is first designed to train a lightweight yet efficient network. We design a deblurring network to search the blurring kernels dynamically, fully exploiting the attention mechanism to focus on the blurring area.

Secondly, universally, image restoration of edge attention is preformed in three steps. First, we abstract the edge information by edge prior, the proposed approach refines the inside features by attention modules to finally reconstruct the whole image. Reconsidering edge attention mechanism for the image prior, we develop a general algorithm for low level image restoration. This method applies a different feature extraction sequence: objects are targeted by a class activated function. Only the main structures and key features of the marked object can be recognized by edge detection. Finally, preset proper kernels are adopted to process the suitable regions.

Thirdly, several techniques are investigated ablation experiments to explore various deep learning strategies. In this study, we verify that image deblurring performs better in joint training than transfer learning or multimodel training. An edge attention algorithm, lightweight residual strategy, fine-tuned weight, and multipath refinement loss function are developed in a plug-and-play architecture to adapt different demands for image processing efficiency, GPU requirements of the model, speed and accuracy balance, and training efficiency. We modify the network in a lightweight manner by combining the iterative and recurrent architectures. The design of a lightweight convolution and residual connection network architecture makes the model more streamlined, efficient, and fast.

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

2. Related Work

2.1. Blurring Kernel Estimation. The early work on image deblurring depended on a variety of assumptions and natural images acquired a priori [19]. Subsequently, some uncertain parameters would be determined in blurring models, such as the type of blurring kernel and additive noise [20, 21]. However, in real applications, these simplified assumptions about sampled scenes and blurring models may lead to performance degradation. In addition, these methods are computationally expensive, and numerous parameters typically need to be adjusted.

In recent years, the application of deep learning and generative networks in computer vision tasks has led to great breakthroughs in many research fields. Several regression networks based on convolution neural network (CNNs) have been proposed for image restoration, including some methods that involve with image deblurring [22, 23]. Compared with traditional methods, the methods based on deep learning are less dependent on prior knowledge. These new models have demonstrated the ablity to reconstruct images more accurately on both global and local scales.

It is generally believed that a blurred image is formed by the convolution of a blurring kernel and additive noise [3]. Therefore, the existing algorithms normally use the blurring kernel function for the deconvolution of a blurred image. The existing algorithms can be divided into two categories, according to whether the blurring kernel is known, including (a) blind image deconvolution (BID) [16, 24, 25] and (b) nonblinded image deconvolution (NBID) [20, 22]. BID

restores a clear image without knowledge of a blurred kernel. It only knows the blurred images it has captured. NBID deblurs images with a known blurring kernel. It is usually difficult to know the blurring kernel in practical applications in advance. Therefore, the requirements of BID are much higher than those of NBID.

Such models may use a known fixed kernel to blur [20, 26]. Recent studies have used end-to-end learning methods to handle the blurring of spatial changes, achieving state-of-the-art performance [23, 27].

Some problems remain with prior deep neural network architecture for image deblurring. First, although neural networks that use deeper architectures are usually effective, it is difficult to explain the impact of individual components in these networks. Second, the evaluation indicators used in image restoration tasks, such as PSNR and SSIM, are usually based on pixel or feature differences between clear natural images and processed images. This tends to improve mathematical similarity rather than the quality of human subjective perception. PSNR measures image quality by calculating the mean square error (MSE). However, there is a gap between the MSE and evaluation performed by a human visual system. SSIM models human visual quality in terms of multiple components, such as brightness, contrast, and structure. These components can be used to assess visual quality, but they are essentially unilateral assessments of the complexity of human vision.

On the assumption of fixed blurring kernels for sensors, we can consider it as a mean blurring operation and can use it to model the blurring estimation as a convolution of a latent image I and blurring kernel k,

$$B = k * I + \alpha, \tag{1}$$

where B and α represent the blurring image and added noise, respectively, and "$*$" is the convolution operator. This is a mathematically ill-posed problem, because different I and k pairs can produce the same B values.

2.2. Attention Mechanism Screening Blurring Kernel. In this study, we reviewed the global average pooling layer proposed in [5] and illustrate how it explicitly enables CNNs to have excellent location capabilities, despite training on image-level tags. Although this technique has been previously proposed as a method for regularization training, we find that it establishes a universally localizable deep representation that can be applied to a variety of tasks. We locate objects with high accuracy even though the global average pool appears simple. Furthermore, we demonstrate that our network can locate differentiated image regions for a variety of tasks, even without training.

The latest work of Zhou et al. [28] shows that the convolution units of each layer of the CNN act as object detectors for the location of objects, even without supervision. This function is lost when classifying objects with fully connected layers. Popular CNNs have recently been proposed to avoid the use of fully connected layers to minimize the number of parameters, while maintaining high performance [5]. To achieve this goal, Lin et al. [5] used global average pooling (GAP) as the structure regulator to prevent overfitting.

It is important to highlight the intuitive difference between GAP and global maximum pooling (GMP). GMP encourages the identification of only one discriminatory part, while GAP encourages the network to identify a range of objects. It is designed to replace fully connected layers in classical CNNs. GMP has been used for weakly supervised object locations in previous research [29]. In our experiments, we found that the advantages of GAP layers extended beyond their functionality as a normalization regulator. With a small adjustment, the network can retain its excellent localization capabilities to the last layer. Distinguishable image areas can be easily identified in a single forward pass using this adjustment to accomplish a variety of tasks, even those for which the network was not initially trained.

The aim is for each unit to be activated by a visual pattern in its receptive area. Therefore, a map of the visual mode is required. The class activation graph is the weighted linear sum of the presence of these visual patterns in different spatial locations. The most relevant images areas to a particular category can be identified by simply sampling the class activation graph to the size of the input image.

Traditional methods rely on blur kernel estimation to reconstruct images by focusing on specific types of blurs [3, 24, 30–32]. Recent studies have attempted to settle the restoration problem by adopting multiscale CNNs to deblur the images. In these end-to-end frameworks, blurry images are used as inputs to the neural network to immediately generate clear images [20]. However, the performance of such methods remains unsatisfactory owing to the fixed assumption of the blurring kernel. CNNs can greatly improve the computational speed of traditional methods, but their prediction accuracy remains inefficient, and they require the use of considerable GPU memory resources.

2.3. Network Architecture. Image deblurring CNNs can be divided into GAN, multiscale, recurrent, multipatch, and scale-iterative architecture networks for feature extraction.

2.3.1. Multiscale Architecture. Multiscale networks [13] extract various features from each scale by scaling an image into different sizes, as shown in Figure 1(a). The input images are converted into feature maps, and scales are used to halve the feature maps at each level. In multiscale detection, the various scale features are fused with different methods and contain a large quantity of information, suggesting the possibility of 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 concatenation processes must wait for the computational results from the small scales, which results in a relatively slow training speed.

2.3.2. Recurrent Architecture. An input layer, loop hiding layer, and output layer constitute a recurrent network [18, 33, 34] as shown in Figure 1(b). Recurrent networks can learn features and long-term dependencies in sequence. However, as the number of network layers increases, so does the

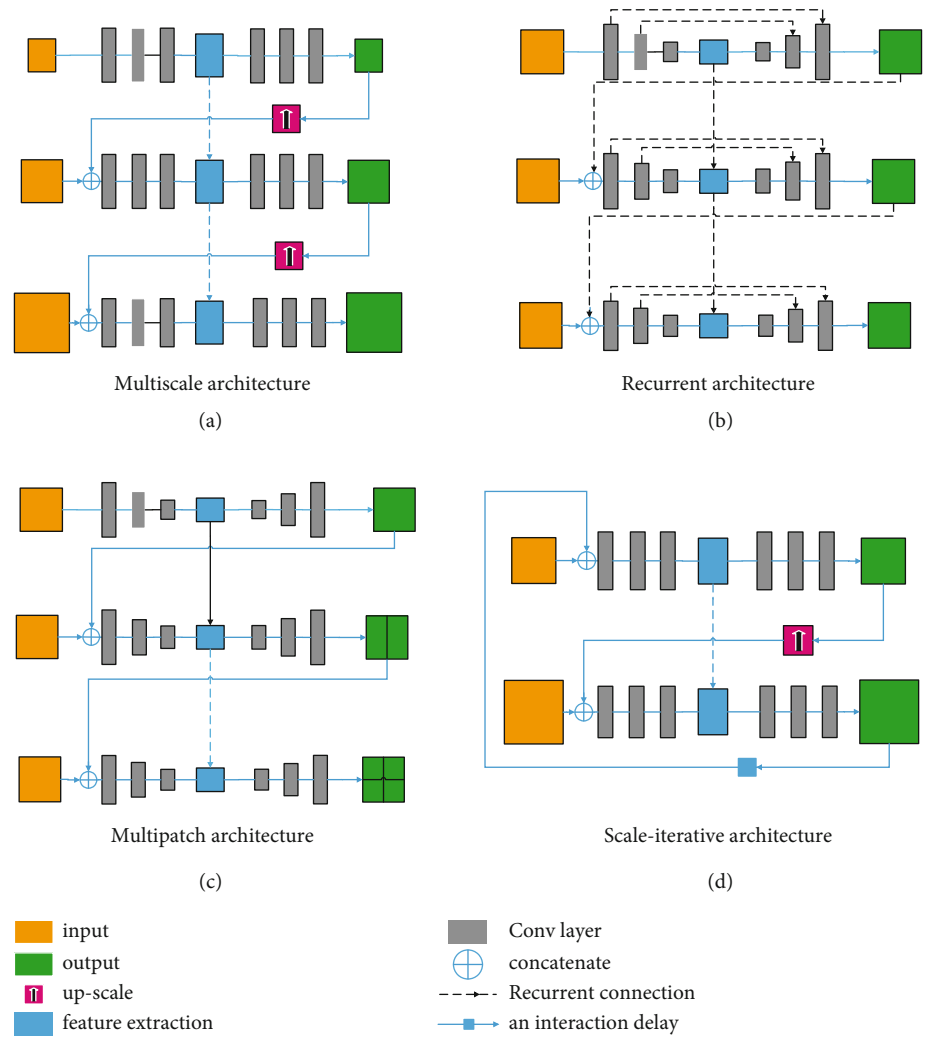

Figure 1: Various deblurring network architectures. (a) Nah et al. [13] proposed the multiscale architecture to extract features from different scales. (b) Tao et al. [15] proposed the recurrent architecture, in which the next round of training can be aided by the last round results. (c) Zhang et al. [17] utilized the multipatch architecture to directly extract features from image pairs by cropping images in different scales. (d) Ye et al. [18] 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.

required computational complexity. The process deteriorates if invalid features are extracted in the last round because the concatenation of recurrent networks relies heavily on last-round results. Subsequently, the deblurring inference becomes extremely unstable if image restorations are of poor quality.

2.3.3. Multipatch Architecture. A deep multipatch hierarchical network (DMPHN) is a CNN model that appears simple but operates as an effective multipatch network, as shown in Figure 1(c) [17]. An input image is divided into different sizes each time. Features were then extracted with the use of a multiscale architecture. Although DMPHN has attained remarkable progress in terms of computational effectiveness, their precision is low.

2.3.4. Iterative Architecture. Ye et al. [18] proposed a scale-iterative upscaling network (SIUN) to iteratively restore sharp images, as shown in Figure 1(d). The super-resolution structure of an upsampling layer was adopted between two consecutive scales to restore the details. Image features are extracted from small to large scales, with the aim of reconstructing high-resolution images from low-resolution originals. The downsampling process begins to restore the image until it is equal to the size of the original image. Moreover, its weight sharing can be preserved, and its training process is flexible. However, the method failed to achieve high deblurring precision and network efficiency, and substantial memory was required for the iterative calculation.

We extend this method by combining the edge feature learning strategy and contextual attention modules for further image restoration, which can locate objects aided by structure information and adopt appropriate deblurring priors to reconstruct sharp images.

3. Model Design and Implementation

The MRFNet is extensively constructed to ensure a balance between accuracy and speed. We first exploit the recurrent and multiscale strategies to learn multifrequency information. A structure is designed with a branch depth and fusion unit on basis of the lightweight process and remote residual connection [35]. Finally, a multiscale refinement loss function is used to train the network in a coarse-to-fine manner.

3.1. Multiscale and Recurrent Learning. The recurrent and multiscale learning strategies are applied in this study. The basic idea of the multiscale learning strategy is to extract features from large coarse scale maps and upsampled results as green lines shown in Figure 2(a). Meanwhile, in the recurrent learning strategy, the high-level feature extraction path acquires fusion information from the low-level refinement maps and the final feedback in the form of purple flow lines, as shown in Figure 2(a). In our study, the two strategies are combined by designing four refinement paths to extract features in different scales, instead of directly predicting the entire deblurred image. Thus, the 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 Figure 2.

In the multipath input stream illustrated in Figure 2(a), the upper MRFNet layer takes blurred and sharp images as input and processes the deblurring datasets in a total of four scales, i.e., k varies from 2 to 4. The four scale blurring 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 upsampling feature maps l_{k+1} into a middle feature map denoted as

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

The fusion unit then adds c_k and the results from the last iteration l_{k-1} to obtain the final outcomes, which is denoted as l_k. This process briefly describes how the refinement fusion path functions. The entire process can be calculated as

$$l_k = c_k + l_{k-1} (2 \le k \le 4). \tag{3}$$

3.2. Lightweight Residual Process. Numbers of parameters and floating-point operations of our original MRF network originate from the commonly used 3×3 convolution. Therefore, we focus on the replacement of these elements with simpler counterparts without compromising performance.

The original design of our MRFNet employs an encoder–decoder structure equipped with four feature extraction and downsampling layers. Each path includes a fusion unit. The basic block uses a 3×3 convolution, which we call the fusion unit. Herein, the 1×1 fusion unit in Figure 3(a) is replaced with a 3×3 convolution. A chained residual pool (CRP) is also considered to naturally illustrate the operation of the lightweight process and how the three former units are reshaped. The lightweight process is applied to the CRP unit by substituting the 5×5 and 3×3 convolutions with the 5×5 and 1×1 convolutions, respectively, as shown in Figure 3(b).

The refinement path adopts a convolution 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. The two convolution layers act as a residual connection unit (RCU). Two RCUs are installed in the encoder, and three RCUs are installed in the decoder. All blocks use 1×1, 3×3, and 1×1 convolutions compared with those in the RCU that use 3×3 and 3×3 convolutions. We call the two convolution layers the lightweight residual connection unit (LWRCU), as illustrated in Figure 3(c).

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 transform the features of each pixel locally from one space to another. Herein, we empirically prove that the replacement with a 1×1 convolution does not weaken the network performance. Specifically, we replaced the 3×3 convolutions in the CRP and fusion block with a 1×1 counterpart. We also modify the RCU to LWRCU with a bottleneck design, as shown in Figure 3(c). This method was able to reduce the number of parameters by more than 50% and the number of triggers by more than 75%, as shown in Table 1. The convolutions have been shown to save considerable computation time without sacrificing performance.

We also enhanced the MRF unit, as illustrated in Figure 3(d). Deep residual networks obtain rich feature information from multisize inputs [36]. Residual blocks, originally derived for image classification tasks,are extensively used to learn robust features and train deeper networks. Residual blocks can address vanishing gradient problems. Thus, we replaced the connection layer in the MRF unit.

Herein, 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 rectifier linear unit 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 given as follows:

$$y = f_2(f_1(x)) + f_4(f_3(f_2(f_1(x)))), \tag{4}$$

where f, x, and y represent the convolution operation, characteristic graph of the input, and 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 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 used the least number of parameters when the number of paths is 3, whereas better performance is achieved when the number of paths was 4. When the number of paths was less than 3, the extracted features were in accurate. When the number of paths exceeds 4, the

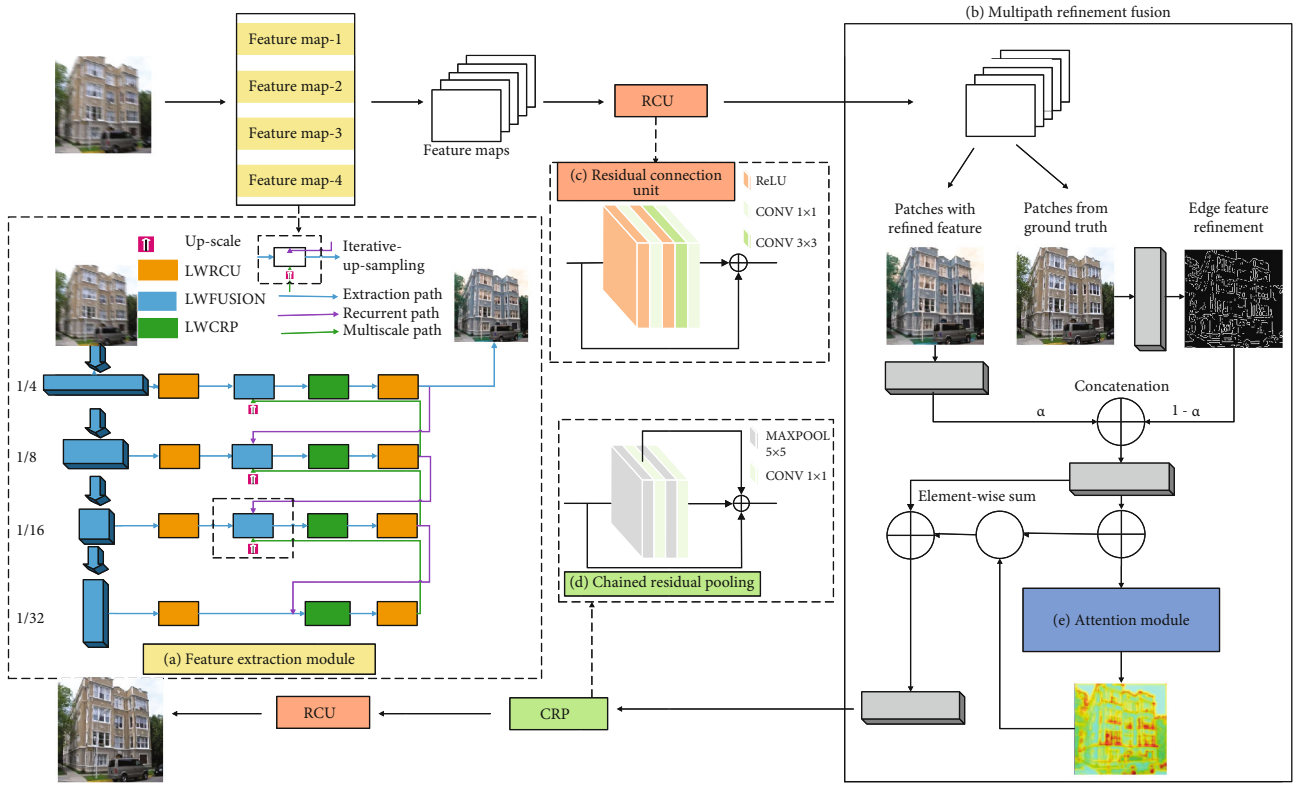

FIGURE 2: MRF framework. The image is separated into different scales from top to bottom. (a) The extraction path of extracting features from scales. (b) Fusion of the recurrent last-round results and the upsampling feature maps as a single refinement process. All four refinement paths finally compute the loss in the scale refinement loss function, and then, the best deblur results are obtained.

deblurring process encountered severe performance degradation, and the training loss remains at a high level continuously. To this end, we chose the four-path refinement setting as the final backbone.

3.3. Loss Design and Training Strategy. Given a pair of sharp and blurred images, MRFNet takes them as input and produces four groups of feature maps at different scales. 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 adopt an 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_i is the ground truth patch, and F is the mapping function generating the restored image from the N-interpolated LR training patches x_l. Herein, 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 includes 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 denotes the k-scale sharp maps. The loss at each scale is normalized by the number of channels c_k, width w_k, and height h_k.

Progressive weighted training process: the entire feature extraction and fusion process is illustrated in Figure 2(b). In the multipath refinement extraction and fusion stages, the task is to fuse the deblurring feature and edge feature from the outputs to generate the final restored frame. The patches with blurry and refined features and the ground truth are input during the training process.

First, the edge feature is extracted from the ground truth patches and the hyper parameter α is initially set to 0 to control the proportion of the refined resource. Second, the refined and mixed edge feature patches are fused in the contextual attention module, which uses the softmax function to predict the foreground and generate the preliminary activated heatmaps. Third, α is set to 1, and the deblurred, refined feature patches are sent to the attention module in the middle of the training process and are then predicted again by the attention module. The results are compared with the synthesis loss function between the predicted deblurring

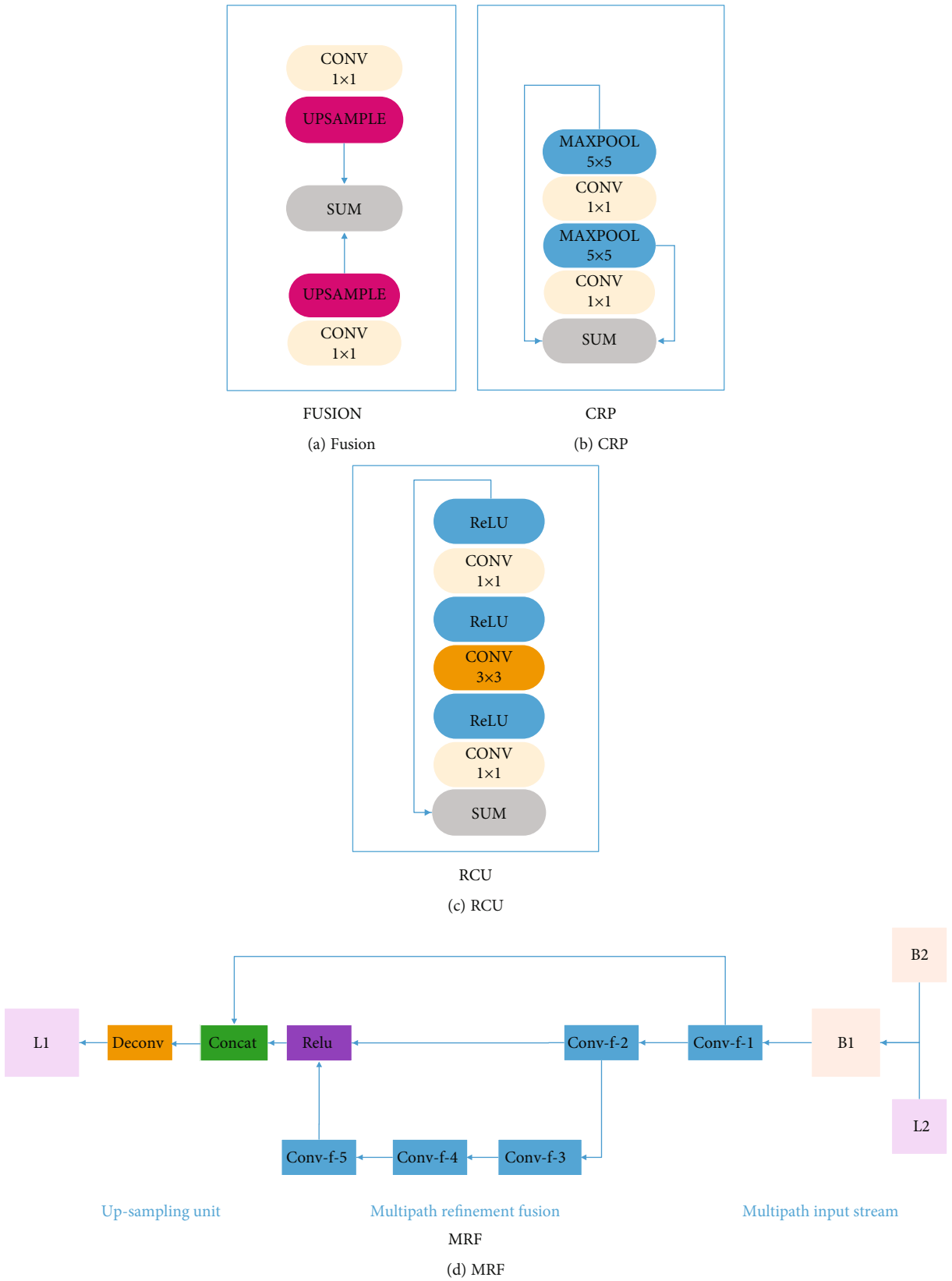

Figure 3: Details of convolutionxal layers: (a) fusion unit, (b) improved CRP module, (c) lightweight network structure of RCU, and (d) MRF unit.

Table 1: Specific parameters of the MRFNet.

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

results and patches with sharp features. Therefore, the deblurring feature refines the input of blurry images and benefits the edge feature extraction at the beginning of the training. In the middle of the training process, the deblurring and edge features are fused by controlling the parameter α. Finally, each path containing different scales of double feature patches is refined and matched with the use of the multipath context attention module with activated heatmaps to infer the final predictions.

4. Performance Evaluation

In this section, we compare MRFNet to recently adopted methods specifically, DeepDeblur [37], DeblurGAN [9], DeblurGANv2 [10], DMPHN [17], and SIUN [18], in terms of accuracy and time efficiency.

4.1. Experimental Setup. MRFNet was implemented using the Caffe deep learning framework. The model was trained with the Adam optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$). Input images were randomly cropped to 256×256 in the training process. A batch size of 16 was used for the training, which are performed with four NVIDIA RTX2080Ti graphical processing GPUs. At the beginning of each epoch, the learning rate was initialized to 10^{-4} and was subsequently halved every 10 epochs. We trained for 170 epochs on the VisDrone dataset and 150 epochs on GOPRO.

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

4.2. Dataset. We used two popular benchmark datasets to train and evaluate the performance of MRFNet: VisDrone and GOPRO. VisDrone provides synthetic blurring techniques and collects real blurry aerial scenarios [38]. GOPRO captures real-world motion blurring scenarios [9]. The images collected from GOPRO were 1280×768, while those of VisDrone were 256×256. The VisDrone dataset included extreme blurry and distorted texture augmentation.

4.3. Comparative Experiments. We conducted comparative experiments using on DeepDeblur [37], DeblurGAN [9], DeblurGANv2 [10], DMPHN [17], and SIUN [18] to verify the performance of our proposed model. The visual effects of different methods are illustrated in Figure 4. MRFNet achieved state-of-the-art performance compared with SIUN and demonstrated clear object boundaries without artifacts Figure 5. The PSNR and SSIM values for MRFNet were much higher than those for DeblurGAN, DeepDeblur, and DMPHN.

Moreover, our method performed better than SIUN and DMPHN and much better than DeblurGANv2 in addressing the GOPRO motion blurs. The trends in Table 2 prove the superiority of the MRFNet framework based on the PSNR and SSIM values. Other methods show considerable limitations in SSIM, indicating that they lack the capacity to restore missing significant structural information and perform deblurring on extremely blurry images.

$$\begin{aligned} \text{claim} &: n_{\text{MRF}} > n_{\text{mean}}, \\ H_0 &: n_{\text{MRF}} \leq n_{\text{mean}}, \\ H_1 &: n_{\text{MRF}} > n_{\text{mean}}, \\ Z &= \frac{T_1 - ((n_1(n_1 + n_2 + 1))/2)}{\sqrt{(n_1 n_2 (n_1 + n_2 + 1))/12}}. \end{aligned} \tag{7}$$

As for the peak of signal-to-noise ratio (PSNR), we can use the data in Table 2 with the Wilcoxon rank-sum test and a 0.05 significance level to test the claim that the

Figure 4: Visual effects of different methods on GOPRO: (a) blurred image and results of (b) DeblurGAN, (c) DMPHN, (d) SIUN, and (e) 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 and produce various generative edge maps for the discriminator D to judge the realness of the generation. The small zoom-in pictures of (e) show the good visual effect of edge attention prior and dynamic kernel selection.

Table 2: Test results of the blurred image datasets and their peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) values.

Method	GOPRO		VisDrone	
	PSNR	SSIM	PSNR	SSIM
DeepDeblur [37]	29.42370	0.761372	27.14940	0.539367
DeblurGAN [9]	28.22642	0.747912	28.29447	0.609642
DeblurGANv2 [10]	32.19638	0.87114	28.43967	0.614876
DMPHN [17]	34.21846	0.898285	28.54136	0.526301
SIUN [18]	34.46135	0.900913	28.28039	0.543417
MRFNET	*34.63429*	*0.907881*	*29.40845*	*0.862474*

multipath refinement fusion n_{MRF} has a confidence larger than mean value of other methods n_{mean}. H_0 is a hypothesis to resist the confidence zone, while H_1 is for it. Z is the specific computation to decide which is correct. The overall deviation is T_1, and n_1 and n_2 are the number of each sample. Then, the value of Z is 1.732; which is larger than 1.645 so that H_1 is in the confidence zone. In conclusion, the performance of MRFNet is better than others in terms of PSNR.

DeblurGAN required the least amount of GPU memory (equal to 4538 MB), while our proposed method required a slightly higher amount of GPU memory than DeblurGAN in GOPRO, as shown in Table 3. This is because DeblurGAN only adopts the generative network for training, which means the model is unstable and the restored color deviates from expectations, as shown in Figure 4(b). MRFNet required the least amount of GPU memory in the VisDrone dataset for a batch size of 16. The lightweight process reduced the

Table 3: Memory consumption of graphics cards.

Method	GOPRO Network (MB) + batch (8)	VisDrone Network (MB) + batch (16)
DeepDeblur	6311	7930
DeblurGAN	*4538*	6012
DeblurGANv2	6861	8107
DMPHN	6541	7329
SIUN	8399	8561
Our model	5452	*5898*

Table 4: Average time of inferring images.

Method	GOPRO Inference time(s)	Times	VisDrone Inference time(s)	Times
DeepDeblur	2.427	1.04X	2.362	1.13X
Deblur GAN	2.346	1.08X	2.144	1.24X
DeblurGANv2	2.528	1.00X	2.663	1.00X
DMPHN	1.886	1.34X	0.764	3.46X
SIUN	0.684	3.69X	0.357	7.46X
MRFNet	*0.494*	*5.12X*	*0.319*	*8.35X*

Table 5: Quantitative numerical PSNR and SSIM results.

Method	GOPRO		VisDrone	
	PSNR	SSIM	PSNR	SSIM
RefineNet [14]	34.17826	0.894369	28.73991	0.854758
LR-RefineNet	34.21445	0.906998	29.24461	0.860164
EA-RefineNet	34.39430	0.903012	29.03971	0.858601
MRFNet	*34.63429*	*0.907881*	*29.40845*	*0.862474*

number of parameters of the model and contributed to low memory usage.

MRFNet was the fastest method in terms of the time of loading the network model and inferences, as shown in Table 4. The inference was also executed on an NVIDIA RTX2080Ti GPU.

4.4. Ablation Experiments. The original MRF network used as the benchmark is denoted as RefineNet [14]. We added the lightweight and residual connection to the benchmark and denoted it as LR-RefineNet. LR-RefineNet adopts multimodel training strategy. We then added the edge reconstruction and attention modules to the refinement path on RefineNet and denoted this combination as EA-RefineNet. EA-RefineNet adopts the joint training strategy. Finally, we combined the lightweight, residual strategy, and attention modules in the benchmark and denoted combination as MRFNet. MRFNet adopts the joint training strategy as shown in Table 5; the LR-RefineNet and EA-RefineNet performed slightly better than RefineNet. MRFNet achieved the most significant numerical results.

The multiscale refinement loss function takes each subtask as an independent component within a joint task, allowing the training process to converge more rapidly and perform better than other methods. The training losses of other approaches markedly decrease during the first round and then consistently remain at a 6% smooth trend in the following training courses. The MRFNet method, aided by the loss weight scheduling technique, exhibited a dramatic downward trend initially and then remained at approximately 4%. The model accuracy improvements (approximately 10% to 21%) attributed to the multiple rounds of training for the four loss weight groups verified the convergence and advantages of our method's training strategy.

The experimental results indicate that MRFNet could achieve considerable precision. Furthermore, MRFNet executed much more quickly than other deblurring models, such as SIUN and DMPHN. Compared with DeblurGAN and DeblurGANv2, the proposed MRFNet model performed well in terms of the speed (increased by 7.4%) and deblurring quality of images (increased by 4.2%). The GPU memory use remained low owing to the added lightweight process. Our method could also recover more details and achieved relatively high SSIM and PSNR values. Images remained unstable and sometimes contained artifacts and color distortions for other models. Conversely, MRFNet was also uesd to perform image deblurring in a stable manner and resulted in high image sharpness.

4.5. Edge Attention Perception. Real-world image capture cannot avoid blurring. For instance, Figure 6(a) shows cars moving fast on a street, which causes motion blurring. The distance from the lens to the car causes a Gaussian blur. We employed the MRFNet to restore images in three steps, including edge reconstruction, localization of the blurring species, and deblurring of the patches. Edge reconstruction: edge information (high-frequency features) is very important for reconstructing images because a sharper background is beneficial for the refinement of different blurring kernels [35]. The inputs are blur and ground-truth pairs. The edge generative network then predicts the structure of the entire image. Subsequently, the pretrained networks preprocess the edge feature information to ensure that the location and class are associated with the deblurred kernels.

A broad view of edge boundaries is illustrated in Figure 6(b). The ground truth images are then preprocessed into grayscale images for further edge feature extraction and are then sent to the discriminator for the comparable benchmark. The generator produces various generative edge maps for the discriminator D to judge how real the generation is.

$$L_{\text{edge}} = \min_{G_e} \max_{D_e} L_{G_e} = \min_{G_e} \left(a_{\text{adv},1} \max_{D_e} (L_{\text{adv},1}) + a_{\text{FM}} L_{\text{AM}} \right). \tag{8}$$

Blurring category location: the attention mechanism acts in a similar manner toneural cells to focus on interesting elements using broad view [25], classification [39], and location techniques [22]. From Figures 6(e)–6(g), we can conclude that

Figure 5: Edge maps and experimental results. Our restored images show vivid colors and sharp details.

changing the receptive field generates different contextual attention results. When the receptive field is large, objects are perceived in their entirety. When the receptive field is small, each object in the image is perceived and the texture is detailed.

First, we search the background using convolutional layers to create a broad view for latent meaningful objects and extract semantic information through a multipath refinement fusion unit. The second step involved classification. For a given image, $g_l(a, b)$ is the spatial information in the lth layer. G_l then represents the sum of $g_l(a, b)$. Thus, for a specific object class, the input $A_l G_l$ is the input of the softmax function. A is the weight corresponding to class, and it predicts the essential level of G_l. Finally, Q is the output of the softmax function and is denoted as $\exp(S)/\sum_e \exp(S)$.

The score S is defined as follows:

$$S = \frac{\sum A \sum g_l(a, b)}{\sum(a, b) \sum A_l \sum g_l(a, b)}. \tag{9}$$

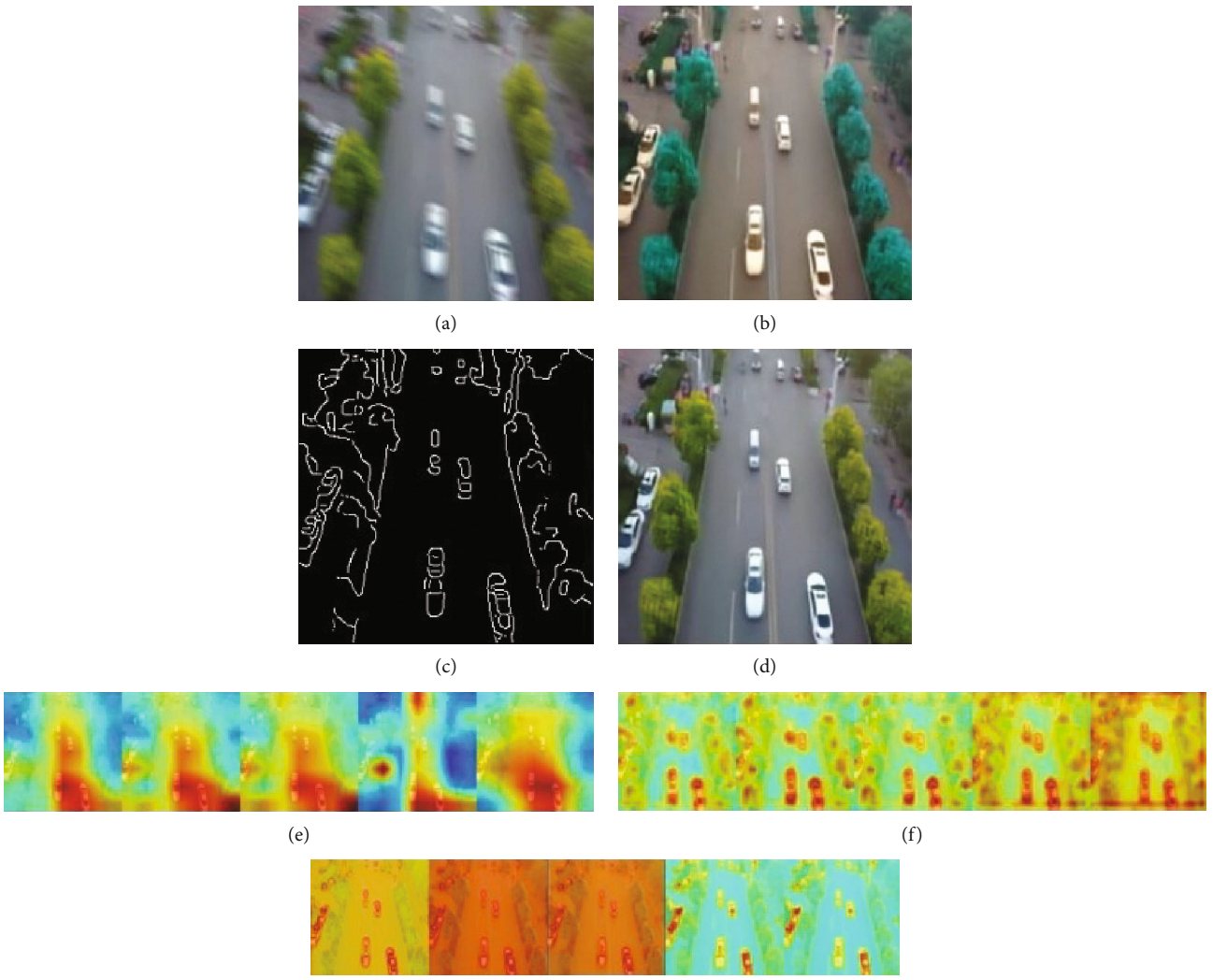

Figure 6: Joint generative image deblurring aided by edge attention prior is used to locate the area which is red in the picture and deblur the patches aided with edge prior. During the process of deblurring, multiple blur kernels are adopted dynamically including motion blur and Gaussian blur and so on. (a) An original blur image. (b) Attention map of specific objects. (c) Receptive field of attention is large, and the scan is with large-scale objects. (f, g) The small receptive fields, respectively. The edge recovery of a generative adversarial network (GAN) tends to be slightly intermittent, but the restoration effect is the best in complex structures. (e–g) The experimental validation of the selective regions to deblur the specific categories of blur objects in multiscale.

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

Third, the deblurring category is located. Based on the edge maps, we can search, locate, and itemize the blurry objects into six categories, including sharp area, random deviation, changeable blur size, changeable shaking angle, changeable shaking length, and motion blurring. In terms of each category, MRFNet uses a different deblurring kernel to refine the blurring features for specific objects. The attention module was able to find and locate the general objects and apply different deblurring approaches through a deep learning training process. Subsequently, the specific objects were deblurred into sharp objects, aided by the edge generation modules and contextual attention mapping.

Patches deblurring: the structure information, predicted object, and blurry potential class could be determined when the data flow from the edge feature extraction and contextual attention were located. Subsequently, we use the deblurring feature prior network to deblur the images into sharper outputs. In this manner, we can restore the image by applying different blurring strategies in various image areas. As a result, the reconstruction of the object structure is meaningful and vivid, and the target is more specific, which improves performance.

5. Conclusions and Future Work

In conclusion, neither edge attention prior nor multimodel training can focus on the core objects in the foreground

and select the proper kernels to restore. Therefore, we have designed a new algorithm consisting of three steps, including focusing, locating, and processing. The key insight of the network model and this algorithm is that the restoration of the key objects can significantly enhance the visual effects of the whole picture and retain the most semantic information. In addition, due to the selection of deblurring part of the regions with the appropriate deblur kernels rather than the whole image efficiently, the accuracy and speed are both optimized to a new level.

This study has illustrated an efficient and accurate joint edge and deblurring GAN for multifrequency feature extraction and fusion called MRFNet. This image deblurring framework uses a generative edge prior and dynamically selects proper deblurring kernels. The model is designed to overcome the challenges posed by the substantial computational resources required by CNNs and poor restoration results obtained with other methods that deal with large-scale datasets or neglect edges and color reconstruction. The proposed model has three main features for processing multiple image tasks, including color, position, and differences. Edge detectors and attention modules are then aggregated into units to refine and learn knowledge. Finally, efficient multilearning features transform a fusion into a final perceptive result.

The proposed network exploits a lightweight process, remote residual connection, edge attention mechanism, and scale refinement loss function to handle real blurring scenarios, preserving fast inference speed and high precision. It can extract different features by scheduling the weight of joint training losses and produce a fusion guided by attention modules. This leads to an efficient image restoration. The proposed MRFNet model was compared with existing models on two popular datasets for deblurring. It achieved state-of-the-art performance compared with other methods on the benchmark datasets.

In the future, we will develop a faster MRFNet model for edge computing devices. The computational capability will likely be much higher than that of the GPUs used in our experiments. The techniques of model compression, including pruning and quantization, will also be explored. This model will also be applied to video deblurring or deblurring of inpainting results at the postprocessing stage.

Data Availability

The authors declare that all data presented in this work were generated during the work and any other source has been appropriately referenced within the manuscript.

Conflicts of Interest

There are no conflicts of interests with any affiliation or person.

Acknowledgments

This research was supported by the Postgraduate Scientific Research Innovation Project of Hunan Province (Number CX20200043).

References

[1] X. Chen, Y. Zhu, W. Liu, J. Sun, and Y. Zhang, "Blur kernel estimation of noisy-blurred image via dynamic structure prior," *Neurocomputing*, vol. 403, pp. 268–281, 2020.

[2] Q. Qi, J. Guo, and W. Jin, "EGAN: non-uniform image deblurring based on edge adversarial mechanism and partial weight sharing network," *Signal Processing: Image Communication*, vol. 88, p. 115952, 2020.

[3] J. Rabbi, N. Ray, M. Schubert, S. Chowdhury, and D. Chao, "Small-object detection in remote sensing images with end-to-end edge-enhanced GAN and object detector network," *Remote Sensing*, vol. 12, no. 9, p. 1432, 2020.

[4] Y. Zhang, Y. Shi, L. Ma, J. Wu, L. Wang, and H. Hong, "Blind natural image deblurring with edge preservation based on L0-regularized gradient prior," *Optik*, vol. 225, p. 165735, 2021.

[5] Z. Fu, Y. Zheng, H. Ye, Y. Kong, J. Yang, and L. He, "Edge-aware deep image deblurring," 2019, https://arxiv.org/abs/1907.02282.

[6] L. Xu, J. S. Ren, C. Liu, and J. Jia, "Deep convolutional neural network for image deconvolution," *Advances in neural information processing systems*, vol. 27, pp. 1790–1798, 2014.

[7] C. J. Schuler, H. Christopher Burger, S. Harmeling, and B. Scholkopf, "A machine learning approach for non-blind image deconvolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1067–1074, Portland, Oregon, 2013.

[8] J. Zhang, J. Pan, W.-S. Lai, R. W. Lau, and M.-H. Yang, "Learning fully convolutional networks for iterative non-blind deconvolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3817–3825, Honolulu, Hawaii, USA, 2017.

[9] O. Kupyn, V. Budzan, M. Mykhailych, D. Mishkin, and J. Matas, "Deblur GAN: blind motion deblurring using conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8183–8192, Salt Lake City, Utah, U. S, 2018.

[10] O. Kupyn, T. Martyniuk, J. Wu, and Z. Wang, "DeblurGAN-v2: deblurring (orders-of-magnitude) faster and better," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8877–8886, Seoul, Korea, 2019.

[11] K. Zhang, W. Zuo, S. Gu, and L. Zhang, "Learning deep CNN denoiser prior for image restoration," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3929–3938, Honolulu, Hawaii, USA, 2017.

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

[13] S. Nah, T. Hyun Kim, and K. Mu Lee, "Deep multi-scale convolutional neural network for dynamic scene deblurring," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3883–3891, Honolulu, Hawaii, USA, 2017.

[14] G. Lin, A. Milan, C. Shen, and I. Reid, "RefineNet: multi-path refinement networks for high-resolution semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5168–5177, Honolulu, Hawaii, USA, 2017.

[15] X. Tao, H. Gao, X. Shen, J. Wang, and J. Jia, "Scale-recurrent network for deep image deblurring," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8174–8182, SALT LAKE CITY, 2018.

[16] V. Nekrasov, C. Shen, and I. Reid, "Light-weight RefineNet for real-time semantic segmentation," 2018, https://arxiv.org/abs/1810.03272.

[17] H. Zhang, Y. Dai, H. Li, and P. Koniusz, "Deep stacked hierarchical multi-patch network for image deblurring," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5978–5986, Long Beach, California, USA, 2019.

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

[19] Q. Shan, J. Jia, and A. Agarwala, "High-quality motion deblurring from a single image," *ACM Transactions on Graphics*, vol. 27, no. 3, pp. 1–10, 2008.

[20] D. Krishnan, T. Tay, and R. Fergus, "Blind deconvolution using a normalized sparsity measure," in *CVPR 2011*, pp. 233–240, Colorado Springs, CO, USA, 2011.

[21] L. Xu and J. Jia, "Two-phase kernel estimation for robust motion deblurring," in *European conference on computer vision*, pp. 157–170, Berlin, Heidelberg, 2010.

[22] Z. Hu and M. H. Yang, "Learning good regions to deblur images," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 345–362, 2015.

[23] Y. Fang and T. Zeng, "Learning deep edge prior for image denoising," *Computer Vision and Image Understanding*, vol. 200, p. 103044, 2020.

[24] J. Pan, Z. Hu, Z. Su, and M.-H. Yang, "Deblurring text images via L0-regularized intensity and gradient prior," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2901–2908, Columbus, Ohio, 2014.

[25] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*, pp. 818–833, Zürich, 2014.

[26] Q. Feng, H. Fei, and W. Wencheng, "Blind image deblurring with reinforced use of edges," *The Visual Computer*, vol. 35, no. 6-8, pp. 1081–1090, 2019.

[27] T. A. Javaran, H. Hassanpour, and V. Abolghasemi, "Non-blind image deconvolution using a regularization based on re-blurring process," *Computer Vision and Image Understanding*, vol. 154, pp. 16–34, 2017.

[28] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Object detectors emerge in deep scene CNNs," in *International Conference on Learning Representations*, The Hilton San Diego Resort & Spa, 2015.

[29] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Is object localization for free? Weakly-supervised learning with convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, Boston, Massachusetts, 2015.

[30] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce, "Non-uniform deblurring for shaken images," *International Journal of Computer Vision*, vol. 98, no. 2, pp. 168–186, 2012.

[31] M. Hirsch, C. J. Schuler, S. Harmeling, and B. Schölkopf, "Fast removal of non-uniform camera shake," in *2011 International Conference on Computer Vision*, pp. 463–470, Barcelona, Spain, 2011.

[32] O. Whyte, J. Sivic, and A. Zisserman, "Deblurring shaken and partially saturated images," *International Journal of Computer Vision*, vol. 110, no. 2, pp. 185–201, 2014.

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

[34] K. Schelten, S. Nowozin, J. Jancsary, C. Rother, and S. Roth, "Interleaved regression tree field cascades for blind image deconvolution," in *2015 IEEE Winter Conference on Applications of Computer Vision*, pp. 494–501, Waikoloa Beach, Hawaii, USA, 2015.

[35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, Nevada, United States, 2016.

[36] Q. Wang, S. Shi, S. Zheng, K. Zhao, and X. Chu, "FADNet: a fast and accurate network for disparity estimation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 101–107, Paris, France, 2020.

[37] J. Mei, Z. Wu, X. Chen, Y. Qiao, H. Ding, and X. Jiang, "DeepDeblur: text image recovery from blur to sharp," *Multimedia Tools and Applications*, vol. 78, no. 13, pp. 18869–18885, 2019.

[38] P. Zhu, D. Du, L. Wen et al., "Vis Drone-VID 2019: the vision meets drone object detection in video challenge results," in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pp. 227–235, Seoul, Korea, 2019.

[39] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, Nevada, United States, 2016.