



Dynamic selection of proper kernels for image deblurring: a multistrategy design

Zhichao Zhang¹ · Hui Chen¹ · Xiaoqing Yin² · Jinsheng Deng² · Weili Li³

Accepted: 22 December 2021 / Published online: 21 April 2022

© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2022

Abstract

The non-blind deblurring approach can adequately deblur single-blur images by applying a suitable mathematical model. In contrast, it cannot satisfactorily deblur images that have multiple blurs. The blind deblurring approach is able to remove various kinds of blurs from an image. However, because the causes of blur in different regions differ, it is difficult to locate and remove all the blurs accurately and also to recover the fine texture details. Considering these weaknesses and strengths of both approaches, we propose a neural network that dynamically selects suitable blur kernels for deblurring. In the proposed method, the most appropriate kernels are extracted by joint training from multiple datasets that contain specific types of blurs to tackle local and global regions in one image. In addition, to further improve the image restoration quality, we designed an edge-attention mechanism to compensate the edges and structures of specific objects. The results of experiments conducted indicate that the dynamic selection of blur kernels combined with the edge attention algorithm not only improves PSNR and SSIM, but also outperforms state-of-the-art methods.

Keywords Deep learning · Image deblurring · Blur Kernel selection · Attention mechanism · Edge perception · Multifeature fusion

1 Introduction

Blurry images generally result from manmade causes such as camera jitter and out of focus shots and natural causes such as fog, rainy weather, and background noise [1]. Such images can be viewed as resulting from the convolution of a latent image and a blur kernel with additional noise. Existing algorithms applied to deal with such images can be divided into two categories according to whether the blur kernel is known (non-blind image deconvolution (NBID)) or unknown (blind image deconvolution (BID)).

Zhichao Zhang and Hui Chen have contributed equally to this study.

✉ Jinsheng Deng
jsdeng@nudt.edu.cn

✉ Weili Li
weiwei6563@163.com

¹ College of Computer, National University of Defense Technology, Changsha, China

² College of Advanced Interdisciplinary Studies, National University of Defense Technology, Changsha, China

³ College of System Engineering, National University of Defense Technology, Changsha, China

NBID methods [2,3] deblur images with a known blur kernel. They establish a solid foundation based on mathematical analysis and theoretical statistics. However, it is usually difficult to know the blur kernel in practical applications in advance. Further, NBID methods have difficulty handling abnormal and multiple-blur images. Moreover, the parallelism and performance of the approach are not sufficiently stable.

BID methods [4–6] restore images with unknown blur kernels and images with unknown blurs. They analyze and learn the blurred images and can learn from datasets flexibly and in parallel by combining deep learning methods and graphics processing units. However, because they learn features by comparing the pair of blur and sharp images or frames, they may learn irrelevant image attributes such as color distortion and fail to precisely deblur images.

To overcome the above issues, we propose training on specific blur categories so that the image-deblurring learning experience is solidified into a comprehensive training model or checkpoint. During testing using this approach, the model matched the blur kernel for the patches precisely. The strategy of selecting specific kernels for multiple blurs in one image

dynamically solves the challenges of image distortion and the speed–accuracy balance.

In addition, considering that after deblurring the restored image may have space dislocations and intermittent lines in the image structure of real-world objects, we also designed an edge attention algorithm that restores the key points of objects during the deblurring process. An edge attention algorithm is used to focus, locate, and process the specific object to enhance the low-level image restoration performance. Our contributions are as follows:

- We propose a novel multipath edge attention network (MEANet) for image deblurring. In terms of one single image with multiple blurs in different areas, blur kernels are selected to match various regions dynamically to remove blurs precisely. Multiple strategies are adopted to optimize the multipath refinement network to achieve marvelous restoration visual effect quickly and accurately.
- We combine the process of structural reconstruction and attention mechanisms, proposing a novel edge attention algorithm for image deblurring. The aim of the algorithm is to focus on the main aspect of an image, recognizing the category of blur in the image.
- We design an iterative and recurrent network to reduce the model’s occupation of the GPU. Residual and lightweight strategies are adopted to reduce the neural network size. Certain convolutional kernels are replaced from 5×5 and 3×3 to 3×3 and 1×1 , which reduces the number of model parameters.

The remainder of this paper is organized as follows: Section 2 discusses various methods and related work. Section 3 presents the methodology and outlines the implementation of our proposed network, illustrating how we resolve the aforementioned challenges. We discuss our experimental results in Sect. 4 and conclude the paper in Sect. 5.

2 Related work

Image blurring is a common phenomenon in real scenes. In general, the blur causes are complex and vary in some parts of the image. It is difficult to identify the causes of blur because they occur simultaneously. In addition, blurring inversion is a strongly ill-posed problem because the blurred image may correspond with multiple clear images.

2.1 Image deblurring

2.1.1 Cause of blur and imaging principle

Before the introduction of deblurring methods, we need to know the causes of blur images and videos through natural and manmade processes. The natural image or frame may be blurry when the light is dim and bad weather conditions such as fog, rain, and wind exist. In such cases, it is essential to evaluate the restoration quality of deblurring by statistical analysis. Manmade blur generation can be produced by simulating the camera jitter, adding a specific blur kernel, transforming images by mixture noise, and so on. As for manmade causes, for example, different choices in aperture size and focal length can lead to Gaussian blurring. Manmade operational errors, camera jitter, and complex scenes of moving objects can cause various kinds of manmade blurs. Therefore, the natural blur causes are passive, while the manmade blur causes are active.

Equation (1) defines the imaging principle and formulates the image generation process. Equation (2) defines the blur accumulation during the continuous imaging process. The image generation process of the camera sensors can be estimated as:

$$\begin{aligned} \text{Image}(\text{Object1} + \text{Object2}) \\ = \text{Image}(\text{Object1}) + \text{Image}(\text{Object2}). \end{aligned} \quad (1)$$

Object1 and Object2 are the patch content of an image. Regardless of the presence of the patch in the image or frames in videos, blur can accumulate when Object1 overlaps with Object2. Here, B is the blur image and T is the exposure time. $s(t)$ and $s(i)$ denote the single-time image capture. When the camera sensor receives light during exposure, it accumulates a clear image stimulus $s[i]$ at each timestep, resulting in a blurred image B [7]. Therefore, the exposure accumulation of the CCD sensor can be defined as follows:

$$B = g \left(\frac{1}{T} \int_{t=0}^T s(t) dt \right) \cong g \left(\frac{1}{m} \sum_{i=0}^{m-1} s[i] \right). \quad (2)$$

Under the assumption of a fixed blurring kernel for the sensor, we can treat it as a mean blurring operation and can use it to model the blurring process as the convolution of a latent image I and blurring kernel k as

$$B = k * I + A, \quad (3)$$

where B and A represent the blurred 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 results.

Equation (3) is the assumption of restoration from the decomposition of the blur image into a kernel and latent image with noise. Equation (4) is a technique for rough estimation of the blur degree.

$$g(x) = x^{\frac{1}{\gamma}} \tag{4}$$

As is well-known, as regards the nonlinear camera response function (CRF) [8], there is no CRF estimation technique available for blurred images with spatial variations [9]. Therefore, when the basic true CRF is not given, the usual practical method is to approximate the CRF as a gamma curve with $\gamma = 2.2$, because it is the approximate average of the known CRF.

The CRF equation is used for artificial blur simulation on a dataset. For the GOPRO and VisDrone datasets, various specific blur kernels have been added according to Eqs. (2) and (3). Therefore, by correcting the gamma function, we obtain a sharp image according to the observed image, and we can obtain a blurred image by fine-tuning the parameter γ , as displayed in Eq. (4). VisDrone provides synthetic blurring techniques and collects real blurry aerial scenarios [10]. GOPRO captures real-world motion-blurring scenarios [11].

2.1.2 Overview of image deblurring methods

Richardson [12] conducted seminal image restoration research in 1972. Early work on image deblurring depended on the assumption of restoration from the decomposition of blur image into kernel and latent image with noise [13]. Subsequently, some uncertain parameters in the blurring model would be determined, such as the type of blurring kernel and additive noise [2,14]. However, in real applications, these simplified assumptions about sampled scenes and blurring models can lead to performance degradation. In addition, these methods are computationally expensive, and numerous parameters usually need to be adjusted.

In recent years, the application of deep learning and generative networks to computer vision tasks has led to breakthroughs in many areas. Several regression networks based on convolutional neural networks (CNNs) have been proposed for image restoration, including some methods that deal with image deblurring [3]. Compared with traditional methods, the methods based on deep learning are less dependent on prior knowledge. The new models can reconstruct images more accurately at global and local scales.

In general, networks may use a known fixed kernel to deblur [2,15]. Recent studies have used end-to-end learning methods to deal with the blurring of spatial changes and achieved state-of-the-art performance [16]. There are many blind deconvolution methods that estimate a sharp image and PSF using a joint optimization process [5,17,18]. There are also many blind deconvolution methods (with and without

joint optimization) that do not rely on specific PSF models [19]. Recent studies have attempted to solve 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 [2]. However, their performance is not satisfactory owing to the assumption of a fixed blurring kernel. CNNs are much faster than traditional methods, but their prediction accuracy is poor, and a considerable amount of GPU memory is utilized.

2.1.3 Quantitative evaluation on image recovery quality

It is necessary to evaluate the confidence of the restoration quality. Thus, the peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) are introduced to assess the signal sharpness and image similarity with respect to the noise and between the original and reconstructed structure.

PSNR [20] is a widely used image quality objective evaluation index. It evaluates the quality of image restoration by comparing the differences between the corresponding pixels of the image. It can be expressed as follows:

$$MSE = \frac{1}{NM} \sum_{i=1}^N \sum_{j=1}^M |I(i, j) - \hat{I}(i, j)|^2, \tag{5}$$

$$PSNR = 10 * \log_{10} \left(\frac{MAX^2}{MSE} \right), \tag{6}$$

where M and N represent the height and width of the image, respectively, MSE represents the mean-squared error between two images, and MAX represents the largest value $2^n - 1$ (n is the number of bits of an image) in the image. I is the pixel position (i, j) of an image. The lower the value of MSE, the more similarity there is between two images, which means the more details are restored. The PSNR usually takes decibels (dB) as its unit, and the larger the value is, the smaller is the degree of image degradation. Because it analyzes the impact of noise on the image at the pixel level, this value may differ from human subjective assessment, which must be considered.

SSIM [20] measures the performance of image restoration by extracting specific structural information from the image. This effectively makes up for the deficiency of PSNR, which only analyzes the error between image pixels, and is more in line with human subjective assessments. It mainly compares the structural similarity of the two images, that is, the degree of image distortion is analyzed with respect to three factors: image brightness, image contrast, and image structure information.

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\mu_x^2 + \mu_y^2 + c_2)} \tag{7}$$

Here, μ_x is the average value of the pixels of image x , μ_y is the average value of the pixels of image y , x is the variance of the pixels of image X , y is the variance of the pixels of image Y , and σ_{xy} is the covariance of image X and image Y . The higher the value is, the lower is the distortion in the image. When two images are exactly the same, the value is one. SSIM uses the mean value of the image to estimate brightness, the standard deviation of the image to estimate contrast, and the covariance of the two images to estimate similarity.

In this paper, the two quantitative indicators PSNR and SSIM are used to evaluate the overall performance of the analysis model for deblurring. To ensure the authenticity and accuracy of the experimental results, we selected the PSNR and SSIM values to analyze the proposed model on multiple datasets.

2.2 Structure reconstruction

2.2.1 Edge detection

In this study, we concentrate on the generation of adversarial edge reconstruction for the overall structure and extract different blurring features. Thus, an inferior performance would be achieved if we did not analyze the specific blurring reasons for a specific image. Even a fast deblurring method does not deblur the entire image because it is not the best option [1,21]. Considering the computational costs, it is also better to have an alternative kernel to deblur different objects at different semantic levels [21]. For better image restoration quality, it is beneficial to combine the blurring category location and important structural information in terms of specific artifacts and degree of blurring.

Regarding edge detection, the Canny edge detector [22] performs well in local edge reconstruction, whereas nested edge detection [23] works well in global reconstruction. The edge recovery of a generative adversarial network (GAN) tends to be slightly intermittent, but its restoration performance is very good for complex structures. The edge GAN (EGAN) exhibits promising performance for edge restoration and image deblurring [24,25]. However, the restoration introduces artifacts if the blurring area has uniform intensity, because it selects the incorrect region for deblurring. Deep learning approaches have been proposed for handling complicated 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. Xu et al. [26], Schuler et al. [27], and Zhang et al. [28] adopted this two-stage traditional procedure, which is based on the use of an encoder–decoder neural network. However, these methods still adopt the traditional framework, which produces unsatisfactory prediction results.

2.2.2 Attention modules

In this paper, we review the global average pooling layer proposed in [21] and illustrate how it explicitly enables CNNs to have excellent location capabilities, despite being trained on image-level annotations. Although this technique has been previously proposed as a method for regularization in training, we have found that it actually establishes a universally localizable deep representation that can be applied to a variety of tasks. We are able to locate objects with high accuracy. Furthermore, we have proven that our network can locate differentiated image regions for various tasks, even though the networks were not trained for these tasks.

The latest work of Zhou et al. [29] shows that the convolution units of each layer of a CNN act as object detectors for the location of objects, even without supervision. This function will fail when classifying objects with fully connected layers. Popular CNNs have recently been proposed to avoid the use of the fully connected layer to minimize the number of parameters while maintaining high performance. To achieve this goal, Lin et al. [21] used global average pooling (GAP) as the structure regulator to prevent overfitting in the training process.

It is important to highlight the intuitive difference between GAP [21] and global maximum pooling (GMP) [30]. GMP encourages the identification of only one discriminatory part, whereas 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 location in the previous research [30]. In our experiments, we found that the advantage of the GAP layer goes beyond its functionality as a normalization regulator. With a small adjustment, the network can retain its excellent localization capabilities up 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.

Various attention modules have been designed to perceive a meaningful object in an image. Some methods utilize an edge prior for image structure reconstruction. The advantages in edge detection or attention modules are obvious; they can handle the objects in images flexibly and quickly, which is ideal for essential or semantic reconstruction. However, they do not exhibit stable performance, and they often need to be embedded into deep learning models.

2.3 Performance enhancement on network architecture

Recent image deblurring methods heavily rely on deep learning models. There are four branches of network architecture: GAN, multiscale, iterative, recurrent. These methods can solve the general image deblurring problems very well.

2.3.1 GAN architecture

GANs have shown promising performance in image deblurring. Kupyń et al. [11] designed a new framework for deblurring that calculates the differences between the generated and original images. Researchers have also achieved substantial improvements using other sophisticated GANs such as DeblurGAN [11], DeblurGANv2 [34], and EGAN [25]. However, a GAN requires a large amount of computing and memory resources when comparing the generated and real images of the discriminator.

2.3.2 Multiscale architecture

Multiscale networks [31] can extract various features from each scale by scaling an image to different sizes, as shown in Fig. 1a. The input images are converted into feature maps, and the sizes of the feature maps are halved at the next level. In multiscale detection[35], the various scale features are fused with different methods and contain a large quantity of information, thus suggesting the results should have high accuracy. However, the multiscale strategy strictly requires

the features to be extracted from small to large scale, which means that large-scale concatenation cannot occur until the computational results from the small scales are available, which reduces the training speed.

2.3.3 Recurrent architecture

An input layer, loop hiding layer, and output layer constitute a recurrent network [32,35,36], as shown in Fig. 1b. Recurrent networks can learn features and long-term dependencies in sequential data. However, as the number of network layers increases, so does the network’s complexity. The process deteriorates if invalid features are extracted in the last iteration because the concatenation of recurrent networks relies heavily on the results from the last iteration. Subsequently, the deblurring inference becomes extremely unstable if image restorations are poor in quality.

2.3.4 Iterative architecture

Ye et al. [33] proposed the scale-iterative upscaling network (SIUN) that iteratively restores sharp images, as shown in

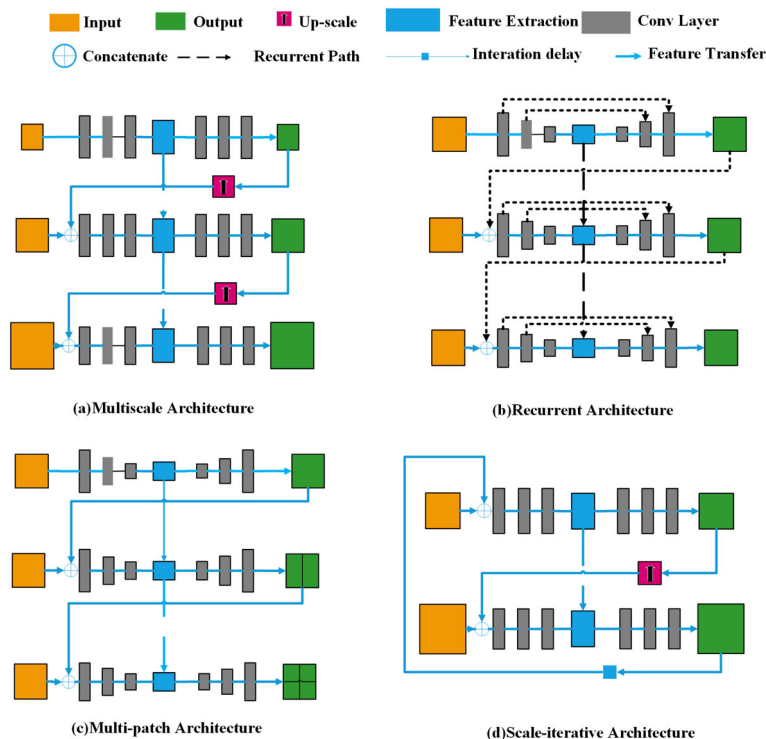


Fig. 1 Various deblurring network architectures. **a** Multiscale architecture for extracting features from different scales [31]. **b** Recurrent architecture in which the next round of training is aided by the results from the previous round [32]. **c** Multipatch architecture for directly extracting features from image pairs by cropping images at different scales [28]. **d** Scale-iterative architecture for training the model with

an upsampling path with the aid of the intermediate results from the last iteration [33]. We combine the ideas of (a) and (b) and propose a new framework called MEANet whose core module uses the multipath refinement MEA. MEANet can operate at both multiple scales and in a recurrent manner

Fig. 1d. The super-resolution structure of an upsampling layer is adopted between two consecutive scales to restore image details. Image features are extracted from small to large scales, with the aim of reconstructing high-resolution images from low-resolution ones. The downsampling process iteratively restores the image until it is equal to the size of the original image. Moreover, the network's weight sharing can be preserved, and its training process is flexible. A deep multipatch hierarchical network (DMPHN) is a CNN model that appears to be a simple network but operates as an effective multipatch network, as shown in Fig. 1c [28]. An input image is divided into different sizes at each iteration. Features are then extracted using a multiscale architecture. However, the method fails to achieve high deblurring precision and network efficiency, and substantial memory is needed for the iterative calculations. However, compared to the attention modules that focus on the main area, these methods do not perform well as part of a large model and thus yield low efficiency.

3 Model design and implementation

The proposed multipath edge attention network (MEANet) in Fig. 2 is constructed to ensure that accuracy and speed are balanced. The backbone of the original MEA network is RefineNet [37]. The blur kernels are selected to match for different regions of a single image dynamically to remove blurs precisely. It is implemented by the edge-attention algorithm in Sect. 3.1. We exploit recurrent and multiscale strategies

to learn multifeature information in Sect. 3.2. A structure is designed with a branch depth and fusion module on the basis of a lightweight process [4] and remote residual connections [9]. Finally, a multiscale refinement loss function is used to train the network in a coarse-to-fine manner in Sect. 4.2. MEANet has a modular architecture for multiple attention modules. It is also modular for several edge detection networks for image information priors and feature extraction, and multiple attention modules can also be added in the multiscale dataflow path. An iterative and recurrent strategy is first designed to train a lightweight yet efficient network.

3.1 Edge attention algorithm

Blurring cannot be avoided in real-world image capture. For instance, Fig. 3a shows fast cars moving on the street, which causes motion blurring. The distance from the lens to the car causes a defocus blur. MEANet restores images in three steps—edge reconstruction, localization of the blurring category, and deblurring of the patches—and suitable preset kernels are adopted to process the good regions.

3.1.1 Edge reconstruction

Edge information (high-frequency features) is very important for reconstructing images because a sharper background helps to refine different blurring kernels [9]. 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 informa-

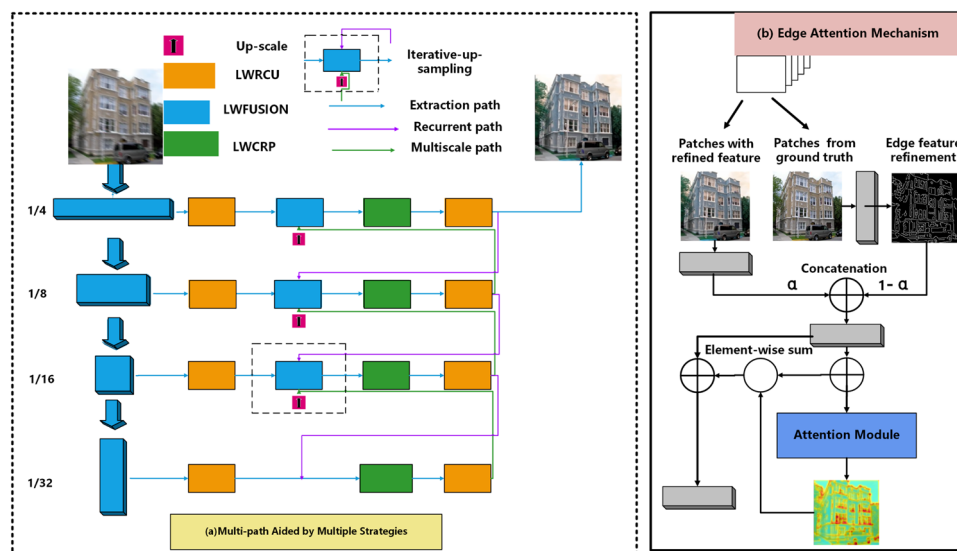


Fig. 2 MEA framework. The input image is augmented into different scales from top to bottom. **a** Path for extracting features at different scales. **b** Fusion of the recurrent last-round results and the upsampling

feature maps as a single refinement process. All four refinement paths compute the final loss in the scale refinement loss function, and then, the best deblur results are obtained

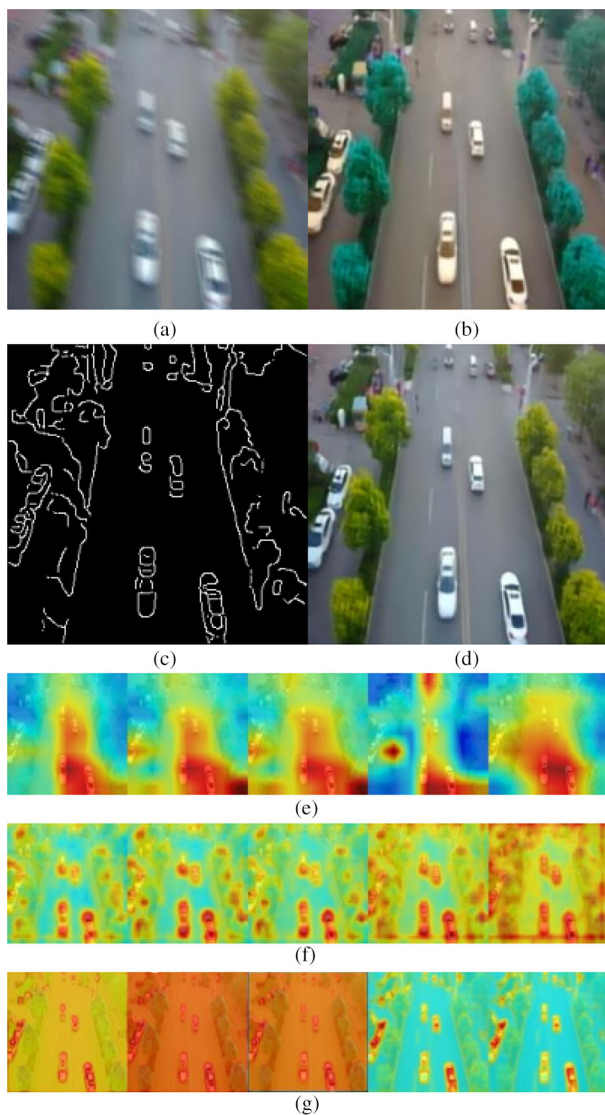


Fig. 3 Multifeature extraction for edge and sharpness. Neither edge attention priors nor multimodel training can focus on the core objects in the foreground and select the correct kernels for restoration. An attention mechanism acts in a similar manner to biological neural cells, and it focuses on objects of interest using a broad viewpoint [6], classification [38], and location [3]. Therefore, we design a new algorithm consisting of three steps: focus, location, and processing. The idea underlying the proposed algorithm is that the restoration of the key objects can significantly enhance the visual effect of the whole image and retain the most important semantic information. Selecting the correct kernels to process specific regions is better than deblurring the whole image aided solely by just one kernel

tion to ensure that the location and class are associated with the deblurred kernels.

An overview of the edge boundaries is illustrated in Fig. 3b. The ground-truth images are then preprocessed into grayscale images for further edge-feature extraction and are sent to the discriminator for the comparison benchmark. The generator produces various generated edge maps for the dis-

criminator D so that it can evaluate how real the generation is

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

3.1.2 Blurring category location

First, we search the background using convolutional layers to create a broad view for determining the latent meaningful objects and extracting semantic information through a multipath edge attention unit. The second step is classification. For a given image, $g_l(a, b)$ is the spatial information in the l -th layer. G_l then 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, and it predicts the essential level of G_l . Finally, S is the output of the softmax function and is denoted as $\frac{\exp(S)}{\sum_e \exp(S)}$. The score S is defined as follows:

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

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.

Next, the deblurring category is located. Based on the edge maps, we can search for, locate, and itemize the blurry objects into six categories: sharp areas, random deviation, changeable blur size, changeable shaking angle, changeable shaking length, and motion blurring. In terms of each category, MEANet has a different deblurring kernel to refine the blurring features for specific objects. The attention module is able to find and locate the general objects and apply different deblurring approaches through a deep learning training process.

Subsequently, the specific objects are deblurred into sharp objects, aided by the edge generative modules and contextual attention mapping.

From Fig. 3e–g, we can conclude that 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.

3.1.3 Patch deblurring

The structure information, predicted object, and candidate blur class can be determined when the data flow from the edge feature extraction and contextual attention are located. Subsequently, we use the deblurring feature prior network to deblur the images into sharper ones. In this way, we can

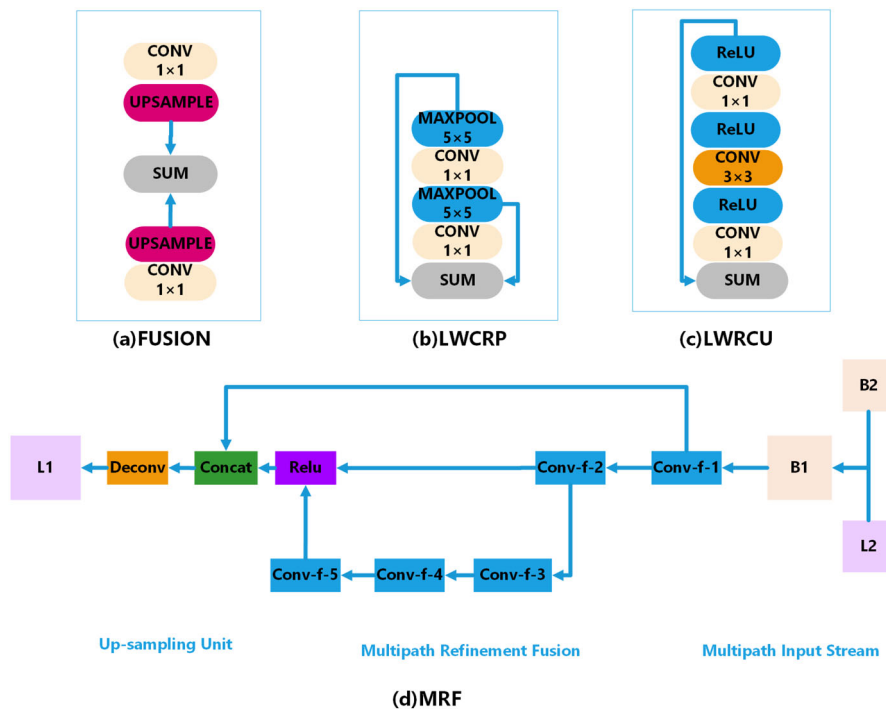


Fig. 4 Different parts of the network. **a** Fusion unit, **b** improved CRP module, **c** lightweight network structure of RCU, and **d** MEA unit. The main purpose for each unit is activation by a visual pattern in its receptive area. Therefore, a map of the visual mode is required. The class activation graph is the weighted linear sum of the existence of

these visual patterns in different spatial locations. The image areas that are most relevant to a particular category can be identified by simply sampling the class activation graph until it is the size of the input image

restore the image by applying different blurring strategies in various image areas. Consequently, the reconstruction of the object structure is meaningful and clear, and the target is more specific, which improves the performance.

The edge attention process can be divided into three steps: abstraction of the edge information by the edge prior, refining the intermediate features by attention modules, and reconstruction of the whole image. We not only obtain the best visual performance, but also enhance the efficiency of patch processing. We adopt an edge generative approach to reconstruct the overall image structure, refining the image from coarse to fine and achieving good performance on a wide range of blurred scenes. The modules satisfy the multilevel requirement of concatenating different types of feature maps and help train an accurate network.

3.2 Multiple strategies

3.2.1 Multiscale and recurrent learning strategies

Multiple strategies are employed in this study. The basic idea of the multiscale learning strategy is to extract features from large- and coarse-scale maps and upsampled results, as shown by the green lines in Fig. 2a. 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, as shown by the purple flow lines in Fig. 2a. 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.

In the multipath input stream illustrated in Figs. 2a and 4d, the upper MEANet layer takes blurred and sharp images as the 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

$$c_k = b_k \oplus l_{k+1} (2 \leq k \leq 4) \tag{10}$$

The fusion unit then adds c_k and the results from the last iteration $l(k-1)$ to obtain the final outcome, which is denoted as l_k . This process briefly describes how the recurrent path

works. The entire process can be calculated as:

$$l_k = c_k + l_{k-1} (2 \leq k \leq 4) \tag{11}$$

3.2.2 Lightweight and residual-connection modification

A large number of parameters and floating-point operations of our original MEANet 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 MEANet employs an encoder–decoder structure equipped with four feature extraction and downsampling layers. Each path has a fusion unit. The basic block uses a 3×3 convolution, which we call the fusion unit. Herein, the 1×1 fusion unit in Fig. 4a is replaced with a 3×3 convolution. A chained residual pool (CRP) is also considered to naturally illustrate how the lightweight process works 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 Fig. 4b.

The refinement path adopts a convolution layer with a stride of one followed by a convolution layer with a stride of two, such that they consistently shrink the feature map size by half. The two convolution layers act as a residual connection unit (RCU) [4]. Two RCUs are installed in the encoder and three in the decoder. All of the blocks use 1×1 , 3×3 , and 1×1 convolutions compared with those in the RCU that use 3×3 and 3×3 convolutions. We call the two convolution layers the lightweight residual connection unit (LWRCU), as illustrated in Fig. 4c.

Intuitively, a convolution with a relatively large core size is designed to increase the size of the receptive 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 replace the 3×3 convolutions in the CRP and fusion block with their 1×1 counterpart. We also modify the RCU to LWRCU with a bottleneck design, as shown in Fig. 4c. This reduces 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 MEA unit illustrated in Fig. 4d. Deep residual networks obtain rich feature information from multisize inputs. The residual block originally derived for the image classification tasks is extensively used to learn robust features and train deeper networks. Residual blocks can address vanishing gradient problems. Thus, we replaced the connection layer with the MEA unit.

Table 1 Specific parameters of the proposed multipath edge attention network (MEANet)

Network	Kernel	Stride	Padding
Conv1	$5 \times 5 \times 32$	1	2
Conv2	$1 \times 1 \times 64$	1	1
Conv3	$5 \times 5 \times 128$	2	2
Conv4	$1 \times 1 \times 128$	1	1
Conv5	$3 \times 3 \times 256$	1	2
Conv6	$1 \times 1 \times 256$	1	1
Conv7	$3 \times 3 \times 256$	1	2
Conv8	$1 \times 1 \times 256$	1	1
Conv_r1_1	$3 \times 3 \times 256$	1	1
Conv_r1_m1	$3 \times 3 \times 256$	1	1
Conv_r1_m2	$3 \times 1 \times 256$	1	1
Conv_r1_m3	$3 \times 3 \times 256$	1	1
Conv_r1_m4	$3 \times 1 \times 256$	1	1
Deconv1	$4 \times 4 \times 128$	1	2
Conv_r2_1	$3 \times 3 \times 128$	1	1
Conv_r2_m2	$1 \times 1 \times 128$	1	1
Conv_r2_m3	$3 \times 3 \times 128$	1	1
Conv_r2_m4	$1 \times 1 \times 128$	1	1
Deconv2	$4 \times 4 \times 64$	1	2
Conv_r3_1	$3 \times 3 \times 64$	1	1
Conv_r3_m1	$3 \times 3 \times 64$	1	1
Conv_r3_m2	$1 \times 1 \times 64$	1	1
Conv_r3_m3	$3 \times 3 \times 64$	1	1
Conv_r3_m4	$1 \times 1 \times 64$	1	1
Deconv3	$4 \times 4 \times 32$	1	2
Conv_r4_1	$3 \times 3 \times 32$	1	1
Conv_r4_m1	$3 \times 3 \times 32$	1	1
Conv_r4_m2	$3 \times 3 \times 32$	1	1
Conv_r4_m3	$1 \times 1 \times 32$	1	1
Conv_r4_m4	$3 \times 3 \times 32$	1	1

Herein, the MEA 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 rectified 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 is as follows:

$$y = f_2 (f_1(x)) + f_4 (f_3 (f_2 (f_1(x)))) \tag{12}$$

where f , x , and y represent the convolution operation, characteristic graph of the input, and characteristic graph of the output, respectively. We construct a shortcut connection in each path of MEANet. In the process of forward transmis-

sion, the remote connections transmit low-level features, which are used to refine the visual details of the coarse high-level feature maps. The inner connections of the convolutional layers allow the gradients to propagate directly to the early convolution layers, thus contributing to more accurate feature transfer.

We set the number of paths from one to six for the multipath process. The operation uses the least number of parameters when the number of paths is three, whereas best accuracy is achieved when the number of paths is four. When the number of paths is less than three, the extracted features are not accurate. When the number of paths exceeds four, the deblurring encounters severe performance degradation. The training loss remains at a high level all the time. Consequently, we chose the four-path refinement setting as the final backbone.

4 Performance evaluation

In this section, we compare MEANet to recently adopted methods—specifically, DeepDeblur [39], DeblurGAN [11], DeblurGANv2 [34], DMPHN [28], and SIUN [33]—in terms of accuracy and time efficiency.

4.1 Experimental setup

MEANet was implemented using the Caffe deep learning framework. The model was trained with Adam ($\beta_1 = 0.9, \beta_2 = 0.999$). Input images were randomly cropped to 256×256 in the training process. A batch size of 16 was used for the training on four NVIDIA RTX2080Ti GPUs. At the beginning of each epoch, the learning rate was initialized to 10^{-4} and subsequently halved every 10 epochs. We trained 170 epochs for VisDrone and 150 epochs for GOPRO. For the sake of time efficiency, we evaluated the inference time of the existing state-of-the-art CNNs on 11 GB RAM RTX2080Ti GPUs.

We used two benchmark datasets to train and evaluate the performance of MEANet: VisDrone [10] and GOPRO [11]. The image size in the GOPRO dataset is 1280×768 , whereas that in the VisDrone dataset is 256×256 . The training dataset, validation dataset, and testing dataset were divided using a ratio of 7:2:1. The total number of images (besides the data augmentation) was 25,000. Gaussian blur was used to deal with the static blurred images in VisDrone. The dynamic blurred images in GOPRO were processed by shooting the motion scene in the field.

To prevent our network from overfitting, several data enhancement techniques are employed. Of the 24,000 pairs of images, 22,000 pairs were used for training and the rest for testing. We augmented the data in VisDrone by including augmentations with extreme blur, distorted texture, cropping

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

4.2 Loss design and training strategy

Given a pair of sharp and blurred images, MEANet 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 $H/32 \times W/32$. In the training process, we adopted an L_2 loss between the predicted deblurring result and the ground truth, as follows:

$$L_2 = \frac{1}{2N} \sum_{i=1}^N \|x_s^i - F(x_l^i)\|^2 \quad (13)$$

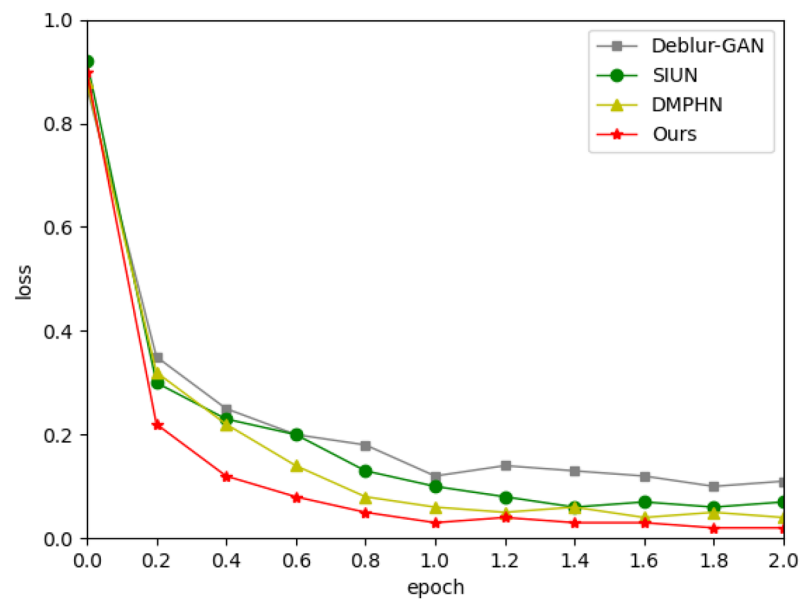
where x_s^i is the ground-truth patch and F is the mapping function that generates the restored image from the N interpolated training patches x_l^i . Herein, the patch size is defined at different levels.

The multiscale refinement loss function is useful for 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:

$$L_{\text{final}} = \frac{1}{2K} \sum_{k=1}^K \frac{1}{C_k W_k H_k} \|L_k - S_k\|^2 + L_{\text{edge}} \quad (14)$$

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 . 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, as displayed in Fig. 5. 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 sessions. The MEANet method, aided by the loss weight scheduling technique, exhibits a dramatic downward trend at first and then

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



remains at approximately 4%. The model accuracy improvements (approximately 10–21%) attributed to the multiple rounds of training for the four loss weight groups verify the convergence and advantages of our method’s training strategy.

4.2.1 Progressive weighted training process

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 hyperparameter 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 one, 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 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 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.

4.3 Comparative experiments

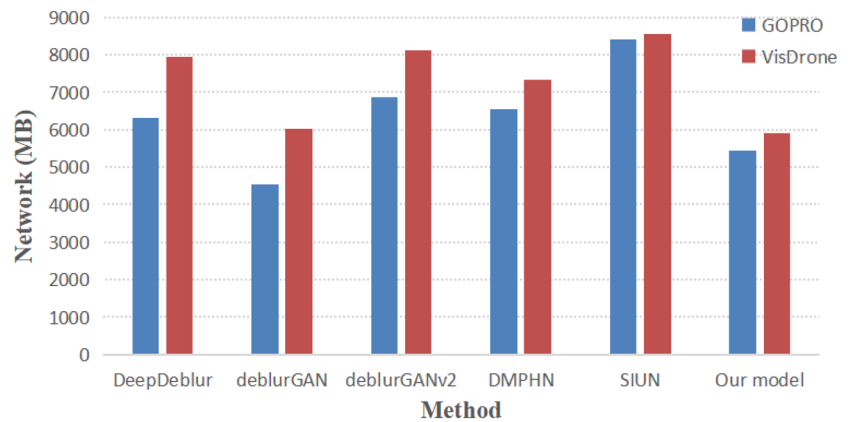
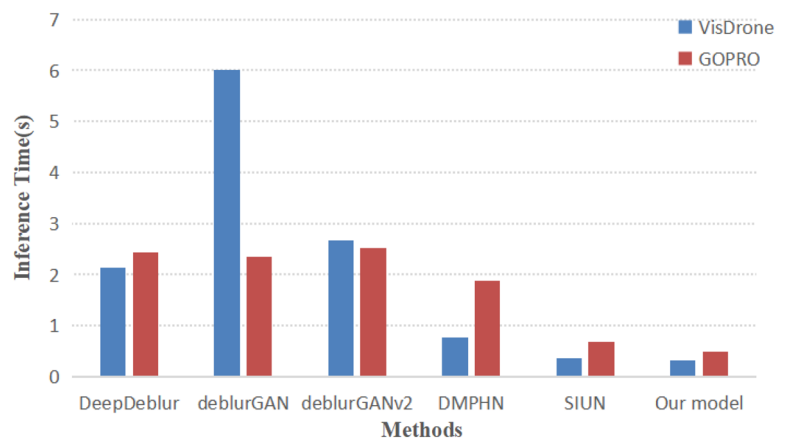
We conducted comparative experiments with DeepDeblur [39], DeblurGAN [11], DeblurGANv2 [34], DMPHN [28], and SIUN [33] to verify the performance of our model. The visual effects of the different methods are shown in Fig. 8. MEANet achieved state-of-the-art performance compared with SIUN and exhibited clear object boundaries without artifacts. The PSNR and SSIM values for MEANet 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 results in Table 2 demonstrate the superiority of the MEANet framework based on the PSNR and SSIM values. Other methods show considerable limitations in SSIM, which means that they lack the capacity to restore a large amount of missing structure information and perform deblurring on images with extreme blur.

DeblurGAN required the least amount of GPU memory (4538 MB), whereas our proposed method required a slightly higher amount for GOPRO, as shown in Fig. 6. This is because DeblurGAN only adopts a generative network for training, which means the model is unstable and the restored color deviates from the expected color, as shown in Fig. 9. MEANet consumed the least amount of GPU memory in the VisDrone dataset for a batch size of 16. The lightweight process reduced the number of parameters of the model and contributed to low memory usage. The average time consumed inferring images is presented in Fig. 7.

Table 2 PSNR and SSIM values obtained on blurry image datasets

Method	GOPRO		VisDrone	
	PSNR	SSIM	PSNR	SSIM
DeepDeblur [39]	29.430280	0.750671	27.150020	0.540512
DeblurGAN [11]	28.225590	0.750112	28.301160	0.611832
DeblurGANv2 [34]	32.204130	0.869930	28.440420	0.615312
DMPHN [28]	34.220210	0.902231	28.538990	0.526201
SIUN [33]	34.458470	0.901142	28.278690	0.544013
MEANet	34.635120	0.908103	29.409320	0.863548

Fig. 6 Memory consumption of graphics cards**Fig. 7** Average time consumed inferring images

4.4 Ablation experiments

In these experiments, the original network benchmark is denoted as RefineNet [37]. We added the lightweight shortcut connection to the benchmark and referred to it as LR-RefineNet. We then added the generative edge reconstruction and attention modules to the refinement path in RefineNet and referred to it as EA-RefineNet. Finally, we combined the lightweight short cut connection, and attention modules in the benchmark and referred to it as MEANet. As shown in Table 3, LR-RefineNet and EA-RefineNet performed slightly better than RefineNet. MEANet has the best numerical results.

Several techniques were applied in the ablation experiments to explore the deep learning strategies. In this paper, we verify that image deblurring performs better in joint training than transfer learning or multimodel training. The edge attention algorithm, lightweight shortcut connection, fine-tuned weight, and multipath refinement loss function were developed to be plug and play to adapt to different demands for image-processing efficiency, GPU occupation of the model, speed and accuracy balance, and training efficiency. We modified the network in a lightweight manner by combining the iterative and recurrent architectures. The design of a lightweight convolution and residual connection makes the model more streamlined, efficient, and fast. Exper-

Table 3 Quantitative numerical PSNR and SSIM results

Method	GOPRO		GPU(MB)
	PSNR	SSIM	
RefineNet [37]	34.178260	0.894369	8399
LR-RefineNet	34.228410	0.910892	6861
EA-RefineNet	34.402120	0.903506	6311
MEANet	34.640220	0.908021	6052
Method	VisDrone		GPU(MB)
	PSNR	SSIM	
RefineNet [37]	28.739910	0.854758	8561
LR-RefineNet	29.252100	0.858932	7930
EA-RefineNet	29.041380	0.860325	7329
MEANet	29.410030	0.863126	6912

iments were conducted to demonstrate the substantial impact of the lightweight process and the residual connection on the enhanced accuracy and decreased complexity of the proposed network. State-of-the-art deblurring performance can be achieved according to the quantitative numerical analysis of the PSNR and SSIM.

The experimental results indicate that MEANet could achieve considerable precision, as shown in Fig. 8. Furthermore, MEANet executed much faster than the other

deblurring models, such as SIUN and DMPHN. Compared with DeblurGAN and DeblurGANv2, the proposed MEANet 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, MEANet performed image deblurring in a stable manner and resulted in high image sharpness.

5 Conclusions and future work

In this paper, we proposed a multipath edge attention network called MEANet deals with the variety of blurs on different regions by dynamically selecting blur kernels. MEANet concentrates on three main challenges in image deblurring: (i) blur kernel estimation for image retrieval; (ii) structure reconstruction and focusing on the main aspects for essential or semantic reconstruction; and (iii) multiple strategies to enhance the efficiency of the neural network.

The network exploits multiple strategies, including a lightweight process, remote residual connection, edge attention mechanism, and scale refinement loss function, to handle

Fig. 8 Results of comparative experiments: **a** input image, **b** ground truth, **c** DeblurGAN, **d** DMPHN, **e** SIUN, and **f** proposed MEANet

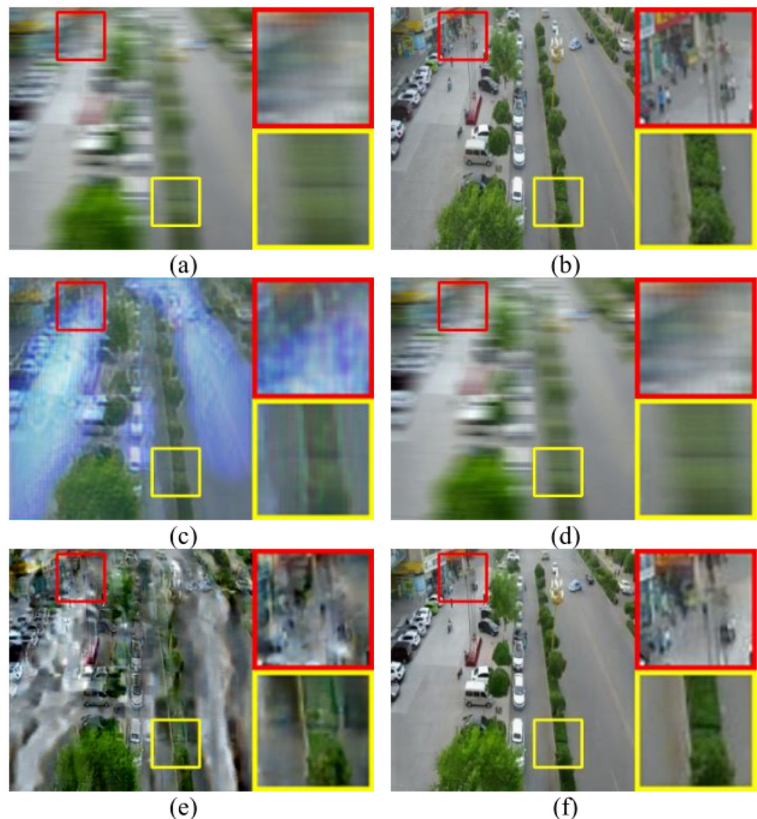
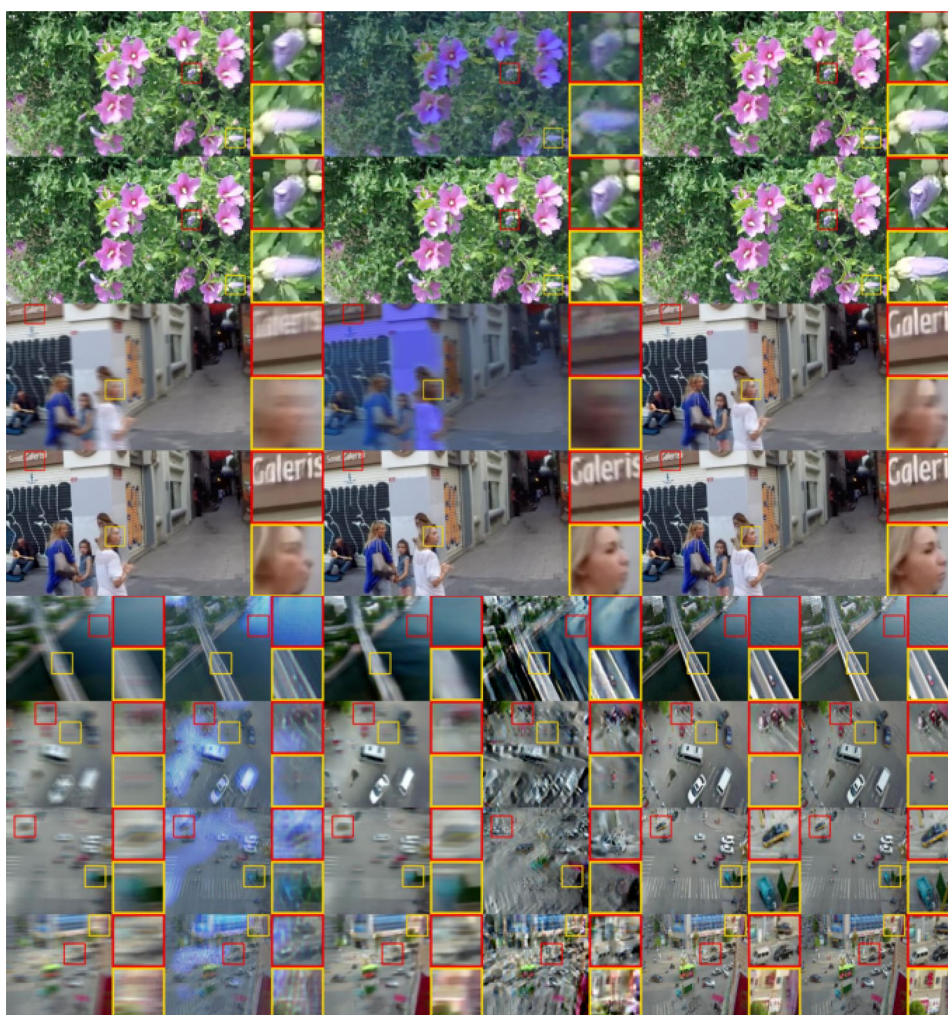


Fig. 9 Results of comparative experiments. The sequence is blurry, DeblurGAN, DMPHN, SIUN, MEANet, and ground-truth images



real blurring scenarios, preserving fast inference speed and high precision. It can extract different features by scheduling the weight of joint training losses and produces a fusion guided by attention modules. This results in efficient image restoration. The proposed MEANet model was compared with existing models on two popular benchmark deblurring datasets. It achieved state-of-the-art performance compared with the other methods on the benchmark datasets.

In future work, we will develop faster deblurring MEANet inferences. The computational capability will likely be much higher than that of the GPUs used in our experiments. Model compression techniques, including pruning and quantization, will also be explored. This model will also be applied to video deblurring or deblurring of inpainting results at the post-processing stage.

Funding This research was supported by the Postgraduate Innovation Project of Double First-class Universities. This work is supported by National Nature Science Foundation of China (grant number:62001493).

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

Code availability The code is available at <https://github.com/zhangzhichao19020123/MEANet>.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

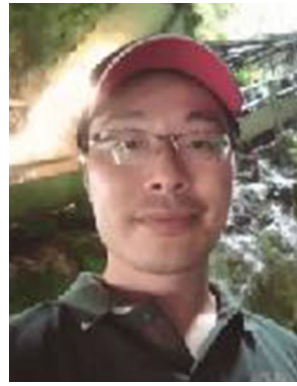
References

1. Li, J., Yang, B., Yang, W., Sun, C., Jianhua, X.: Subspace-based multi-view fusion for instance-level image retrieval. *Vis. Comput.* **37**(3), 619–633 (2021)
2. Krishnan, D., Tay, T., Fergus, R.: Blind deconvolution using a normalized sparsity measure. In: *CVPR 2011*, pp. 233–240. IEEE (2011)
3. Zhe, H., Yang, M.-H.: Learning good regions to deblur images. *Int. J. Comput. Vis.* **115**(3), 345–362 (2015)

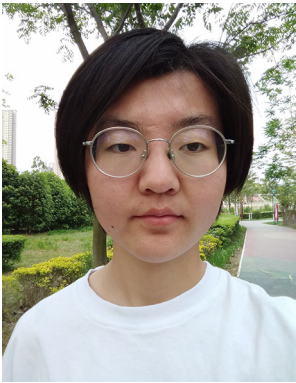
4. Nekrasov, V., Shen, C., Reid, I.: Light-weight refinenet for real-time semantic segmentation. arXiv preprint [arXiv:1810.03272](https://arxiv.org/abs/1810.03272) (2018)
5. Pan, J., Hu, Z., Su, Z., Yang, M.-H.: Deblurring text images via l0-regularized intensity and gradient prior. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2901–2908 (2014)
6. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: European Conference on Computer Vision, pp. 818–833. Springer (2014)
7. Zhang, K., Liang, J., Van Gool, L., Timofte, R.: Designing a practical degradation model for deep blind image super-resolution. arXiv preprint [arXiv:2103.14006](https://arxiv.org/abs/2103.14006) (2021)
8. Shi, W., Huiqian, D., Mei, W., Ma, Z.: (sarn) spatial-wise attention residual network for image super-resolution. *Vis. Comput.* **37**(6), 1569–1580 (2021)
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
10. Zhu, P., Du, D., Wen, L., Bian, X., Ling, H., Hu, Q., Peng, T., Zheng, J., Wang, X., Zhang, Y., et al.: Visdrone-vid2019: 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 (2019)
11. Kupyń, O., Budzan, V., Mykhailych, M., Mishkin, D., Matas, J.: Deblurgan: Blind motion deblurring using conditional adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8183–8192 (2018)
12. Richardson, W.H.: Bayesian-based iterative method of image restoration. *J. Opt. Soc. Am.* **62**(1), 55–59 (1972)
13. Shan, Q., Jia, J., Agarwala, A.: High-quality motion deblurring from a single image. *ACM Trans. Graph. (tog)* **27**(3), 1–10 (2008)
14. Xu, L., Jia, J.: Two-phase kernel estimation for robust motion deblurring. In: European Conference on Computer Vision, pp. 157–170. Springer (2010)
15. Kyrki, V., Kragic, D.: Computer and robot vision [TC spotlight]. *IEEE Robot. Autom. Mag.* **18**(2), 121–122 (2011)
16. Javaran, T.A., Hassanpour, H., Abolghasemi, V.: Non-blind image deconvolution using a regularization based on re-blurring process. *Comput. Vis. Image Underst.* **154**, 16–34 (2017)
17. Hirsch, M., Schuler, C.J., Harmeling, S., Schölkopf, B.: Fast removal of non-uniform camera shake. In: 2011 International Conference on Computer Vision, pp. 463–470. IEEE (2011)
18. Whyte, O., Sivic, J., Zisserman, A.: Deblurring shaken and partially saturated images. *Int. J. Comput. Vis.* **110**(2), 185–201 (2014)
19. Liu, G., Chang, S., Ma, Y.: Blind image deblurring using spectral properties of convolution operators. *IEEE Trans. Image Process.* **23**(12), 5047–5056 (2014)
20. Hore, A., Ziou, D.: Image quality metrics: Psnr vs. ssim. In: *2010 20th International Conference on Pattern Recognition*, pp. 2366–2369. IEEE (2010)
21. Lin, M., Chen, Q., Yan, S.: Network in network. arXiv preprint [arXiv:1312.4400](https://arxiv.org/abs/1312.4400) (2013)
22. Rong, W., Li, Z., Zhang, W., Sun, L.: An improved canny edge detection algorithm. In: 2014 IEEE International Conference on Mechatronics and Automation, pp. 577–582. IEEE (2014)
23. Xie, S., Tu, Z.: Holistically-nested edge detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1395–1403 (2015)
24. Zheng, S., Zhu, Z., Cheng, J., Guo, Y., Zhao, Y.: Edge heuristic GAN for non-uniform blind deblurring. *IEEE Signal Process. Lett.* **26**(10), 1546–1550 (2019)
25. Qi, Q., Guo, J., Jin, W.: Egan: Non-uniform image deblurring based on edge adversarial mechanism and partial weight sharing network. *Signal Process. Image Commun.* **88**, 115952 (2020)
26. Li, X., Ren, J.S., Liu, C., Jia, J.: Deep convolutional neural network for image deconvolution. *Adv. Neural. Inf. Process. Syst.* **27**, 1790–1798 (2014)
27. Schuler, C.J., Burger, H.C., Harmeling, S., Schölkopf, B.: A machine learning approach for non-blind image deconvolution. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1067–1074 (2013)
28. Zhang, H., Dai, Y., Li, H., Koniusz, P.: 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 (2019)
29. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Object detectors emerge in deep scene cnns. arXiv preprint [arXiv:1412.6856](https://arxiv.org/abs/1412.6856) (2014)
30. Oquab, M., Bottou, L., Laptev, I., Sivic, J.: Is object localization for free?-Weakly-supervised learning with convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 685–694 (2015)
31. Nah, S., Kim, T.H., Lee, K.M.: 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 (2017)
32. Tao, X., Gao, H., Shen, X., Wang, J., Jia, J.: Scale-recurrent network for deep image deblurring. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8174–8182 (2018)
33. Ye, M., Lyu, D., Chen, G.: Scale-iterative upscaling network for image deblurring. *IEEE Access* **8**, 18316–18325 (2020)
34. Kupyń, O., Martyniuk, T., Wu, J., Wang, Z.: Deblurgan-v2: Deblurring (orders-of-magnitude) faster and better. In: Proceedings of the IEEE/CVF international conference on computer vision, pp. 8878–8887 (2019)
35. Dai, J., Yang, D., Zhu, T., Wang, Y., Gao, L.: Multiscale residual convolution neural network and sector descriptor-based road detection method. *IEEE Access* **7**, 173377–173392 (2019)
36. Schelten, K., Nowozin, S., Jancsary, J., Rother, C., Roth, S.: Interleaved regression tree field cascades for blind image deconvolution. In: 2015 IEEE Winter Conference on Applications of Computer Vision, pp. 494–501. IEEE (2015)
37. Lin, G., Milan, A., Shen, C., Reid, I.: Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1925–1934 (2017)
38. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Learning deep features for discriminative localization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2921–2929 (2016)
39. Mei, J., Ziming, W., Xiang Chen, Yu., Qiao, H.D., Jiang, X.: Deepdeblur: text image recovery from blur to sharp. *Multimed. Tools Appl.* **78**(13), 18869–18885 (2019)



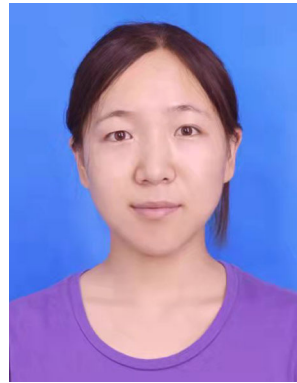
Zhichao Zhang researches in the field of Computer Vision. He is a master student in the College of Computer, National University of Defense Technology.



Jinsheng Deng researches in the field of Data Mining. He is a full professor in the College of Advanced Interdisciplinary Studies, National University of Defense Technology.



Hui Chen researches in the field of Robotics. She is a master student in the College of Computer, National University of Defense Technology.



Weili Li researches in the field of image processing. She is a lecturer in the College of System Engineering, National University of Defense Technology.



Xiaoqing Yin researches in the field of Computer Vision. He is an assistant professor in the College of Advanced Interdisciplinary Studies, National University of Defense Technology.