

Multi-Domain Adaptation for Motion Deblurring

Kai Zhuang, Qiang Li, *Member, IEEE*, Yuan Yuan, *Senior Member, IEEE*, Qi Wang, *Senior Member, IEEE*

Abstract—Motion deblurring is an important topic in the field of image enhancement, which has widespread applications including video surveillance, object detection, etc. Many algorithms are designed for motion deblurring and achieve remarkable performance. However, mainstream motion blur datasets are collected under normal weather and illuminance conditions, i.e., normal domain, ignoring their variations. As a result, current methods perform poorly in dynamic real-world scenes. To address these issues, we study the work in two aspects. First, we collect the real-world motion blur dataset with a well-designed collection device from various angles, focal lengths, and street scenes. Considering its domain is single, it is augmented via a Domain Transfer Strategy (DTS) to construct a Multi-Domain dataset (MD dataset), expanding the domains of the collected dataset. Second, we propose a Multi-Domain Adaptive Deblur Network (MDADNet) with two modules. The one is the Domain Adaptation (DA) module that exploits domain invariant features to stabilize the performance of the MDADNet in multiple domains. The other is the Meta Deblurring (MDB) module that employs the auxiliary branch to enhance the deblurring ability. It also enables the MDADNet to update parameters during the testing stage, improving the generalizations of the MDADNet. Extensive experimental results demonstrate that the MD-trained methods significantly strengthen the motion deblurring ability in multiple domains. Particularly, the proposed MDADNet achieves state-of-the-art performance on the MD dataset and public motion blur datasets.

Index Terms—Motion deblurring, domain transfer strategy, multi-domain dataset, meta deblurring.

I. INTRODUCTION

MOTION blur typically occurs when there exists motion between the camera and the object, causing the loss of details and contour clarity. It not only hinders the enjoyment of the aesthetic qualities of an image but also limits various visual tasks, e.g., blurry images have a large impact on object detection, video surveillance, and public safety tasks. To solve this problem, many algorithms for motion deblurring are proposed.

Currently, numerous methods [1], [2], [3], [4], [5], [6] employ convolutional neural networks (CNNs) to enhance the image quality and achieve outstanding performance compared with conventional methods [7], [8], [9]. These deep learning methods are data-driven and hypothesize that the distribution

This work was supported by the National Natural Science Foundation of China under Grant U21B2041, 61825603, National Key R&D Program of China 2020YFB2103902.

Kai Zhuang is with the School of Computer Science, and with the School of Artificial Intelligence, Optics and Electronics (iOPEN), Northwestern Polytechnical University, Xi'an 710072, Shaanxi, P. R. China. (email: zhuangkai@mail.nwpu.edu.cn)

Qiang Li, Yuan Yuan, and Qi Wang are with the School of Artificial Intelligence, Optics and Electronics (iOPEN), Northwestern Polytechnical University, Xi'an 710072, Shaanxi, P.R. China. (email: liqmgas@gmail.com, y.yuan1.ieee@gmail.com, crabwq@gmail.com)

Qi Wang is the corresponding author.

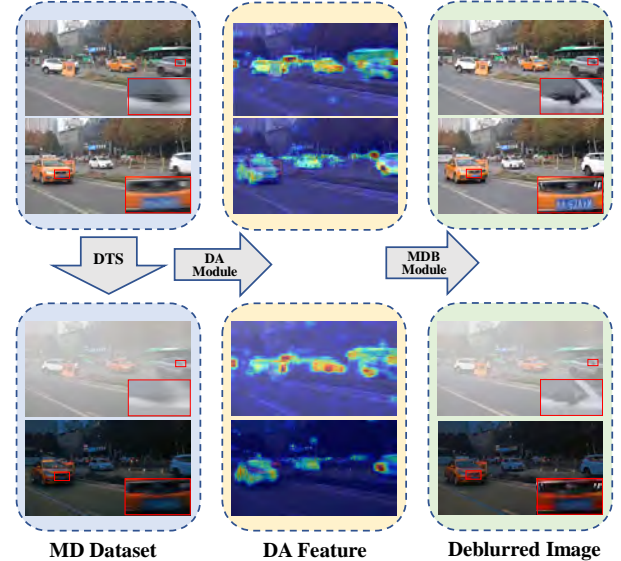


Fig. 1. Visual results of each module for MDADNet. The first row represents the results in the normal domain. The second row shows the results in the foggy domain and low-light domain.

of the training data is identical to the distribution of the real world. However, the data distribution under the motion deblurring task is multivariate in the wild, which can not be adequately represented by a single dataset. In a given dataset, variations in illuminance and weather can lead to inconsistent distribution of the dataset. Since domain typically refers to the source of data, and different data sources may result in different data distributions, these data distributions can be considered as different domains. Hence, the changes in illuminance and weather can be defined as various domains.

Generally, there are two problems in the motion deblurring field: 1) most motion blur datasets are synthetic, which causes their blur kernels to be different from the real world. This gives rise to the degradation of deblurring performance when tested in real-world scenes. To process this problem, some methods [10], [11] design a beamsplitter acquisition system to collect real-world datasets in the wild. However, the construction process for a beamsplitter is rather intricate. It is challenging to move around the beamsplitter to collect datasets that contain adequate scenes. Moreover, the registration of cameras is a highly arduous task; 2) some methods only focus on deblurring performance in the normal domain, ignoring the variations of weather and illuminance, which limits the generalization of the deblurring models. To handle this problem, some methods [12], [13], [14], [15] attempt to restore images in the low-light environment by extracting low-frequency and high-frequency information, which improves the ability of motion deblurring. However, these approaches only work in a single domain.

When tested in other domains or the color between the blurry object and the background is similar, the performance of these methods for motion deblurring is limited.

Considering these problems, a real-world dataset for motion deblurring is constructed in a simple and novel manner: two cameras fixed on the snaplock of the tripod are used to capture image pairs in parallel. For the diversity of the collected dataset, we collect it via the principles of multiple locations, angles, and focal lengths. However, it is impossible to construct a motion blur dataset that contains various scenes in the wild. To alleviate this dilemma, we further expand the scenes via domain transfer strategies (DTS) to construct a multi-domain dataset (MD dataset). Considering that low-light environment, foggy and rainy weather are common in the real world, the normal domain is transferred to the low-light, foggy, and rainy domains at the image level. Notably, in this paper, normal domain image is defined as the image in normal illuminance and weather conditions.

Through the above analysis, in our paper, the multi-domain adaptive deblur network (MDADNet) is proposed to restore blurry images. Different from current deblurring methods, MDADNet could maintain outstanding performance in various illuminance and weather conditions. It mainly consists of two modules, i.e., the domain adaptation (DA) module and meta deblurring (MDB) module, respectively. As for the DA module, it aims to ensure outstanding deblurring performance in multi-domain scenarios by disentangling domain adaptation features from the global feature map. As for the MDB module, it aims to polish the deblurring performance via the auxiliary branch. It complements the primary branch and enables the model to quickly adapt to the image distribution during the testing stage. Fig. 1 shows the results of the DA and MDB module. It can be observed that the DA features focus on blurry objects rather than the background domain, and the deblurred images display the cross-domain ability of the MDADNet. In summary, the main contributions of this work are presented as

- 1) A multi-domain motion blur dataset (MD dataset) is constructed in a novel manner by leveraging domain transfer strategy to expand the normal domain to the low-light, foggy, and rainy domains. Its diversity enhances the performance of the methods trained on MD dataset.
- 2) A Multi-Domain Adaptive Deblur Network (MDADNet) is proposed to boost deblurring performance and improve robustness in multiple domains. Since the auxiliary branch only requires blurry images without their corresponding labels, MDADNet could adjust the parameters of the network in the testing stage. It significantly alleviates performance degradation.
- 3) The results of mainstream methods trained on our constructed MD dataset have significant improvement compared with public datasets. It demonstrates the superiority of the MD dataset. Besides, the MDADNet achieves the best results among the current methods for the motion deblurring task, which presents the excellence of the MDADNet.

The rest of this paper is organized as follows. Section II reviews the related work briefly. Section III describes the creation of the MD dataset. Section IV introduces the proposed MDADNet in detail. The experimental results are displayed and analyzed in Section V. Finally, this work is summarized in Section VI.

II. RELATED WORK

In this section, we review motion deblurring datasets and methods with related work. Besides, the proposed MDADNet employs domain adaptation methods for motion deblurring. Hence, several related works are briefly described.

Motion Blur Datasets. Early motion blur datasets are synthetic. They are constructed in two ways. One [16], [17], [18] is to accumulate clear images in exposure time to generate the corresponding blurry image. To be specific, the GoPro dataset [16] utilizes GoPro Hero Black to collect 3214 image pairs. The REDS dataset [17] totally captures 8422 image pairs, and the Adobe dataset [18] is created by iPhone6s, GoPro Hero Black4, and Canon 7D these three handheld cameras, which contains 6708 image pairs. The other one [19], [20], [21] utilizes synthetic blur kernels to generate blurry images. For example, Sun *et al.* [19] and Xu *et al.* [20] estimate linear blur kernels to convolute with clear images to generate blurry images. Kupyn *et al.* [21] simulate more realistic and complex blur kernels based on random trajectory generation methods.

The blur kernels of these datasets are obtained in a synthetic manner, which is different from the real world. It results in the poor performance of the synthetic dataset trained methods to remove the real-world blur. To handle this, Rim *et al.* [10] and Zhong *et al.* [11] use the beamsplitter to collect the real-world motion blur dataset. However, the beamsplitter is too heavy to move around, and it is hard to conduct mechanical alignment. Moreover, the scenes in the collected dataset only contain static illuminance and weather condition, i.e., the domain of their datasets is single.

Learning-Based Motion Deblurring. With the rapid development of deep learning, numerous data-driven methods [19], [22], [23], [24], [16] have been proposed to process motion blur. Different from model-based motion deblurring methods, deep learning methods are trained on large datasets and achieve outstanding performance. For instance, Sun *et al.* [19] and Gong *et al.* [22] adopt CNN as the backbone to restore blurry images. However, their models are not deep enough to obtain a large receptive field to deblur large blurry areas. To handle it, Tao *et al.* [24] and Nah *et al.* [16] propose a scale-recurrent network to extract coarse-to-fine features. The coarse features capture the structure information of the large-scale blurry objects and the fine features contain the details of the local areas. The integration of coarse-to-fine features is conducive to deblur large blurry areas. Zhang *et al.* [23] further improve the algorithm by encoder-decoder architecture to analyze multi-scale features. It guarantees the deblurring performance in different scale blurry areas.

Recently, Tran *et al.* [25] encode blur kernel space to process various blurry images. Chi *et al.* [26] propose a



Fig. 2. Display of the proposed MD dataset. The first row is the normal domain image. The second row is the transferred domain image.

test-adaptive architecture to adapt quickly during the testing stage, which increments the robustness of the deblurring model. Zamir *et al.* [27] present a transformer-based motion deblurring algorithm to capture long-range pixel interactions. Chen *et al.* [28] propose a simple and computationally efficient method for motion deblurring. However, current deep learning methods are data-driven, which means datasets are crucial to the performance for deblurring. Once the domain gap between the testing data and the training data is large, the performance of deep learning algorithms will degrade.

Domain Adaptation Methods. Domain adaptation aims to enable the model to maintain stable performance when tested in the target domain. Currently, many networks are proposed to study it. Zhu *et al.* [29] propose CycleGAN to capture the feature of the target domain and adapt the feature into the source domain. Sankaranarayanan *et al.* [30] not only use GAN [31] to synthesize images but also utilize GAN [31] to align the distribution of the target domain and source domain. Apart from these image-level domain adaptation methods, some feature-level domain adaptation algorithms are proposed. Different from image-level domain adaptation methods, it aims to extract domain invariant features. Sun *et al.* [32] propose the CORAL to minimize the domain gap between the source domain and target domain without any target labels. Ganin *et al.* [33] first adopt adversarial feature learning in domain adaptation field. The main idea is to use a domain classifier to discriminate the domain category of the feature generated by domain feature extractor. To improve the generalization, Zhang *et al.* [34] pretrain the model with unlabeled data from various source domains. In [35], Kendall *et al.* claim that multi-task learning has a large impact on the performance of domain adaptation methods. Inspired by this work, Mukherjee *et al.* [36], [37] combine the multi-task learning and domain adaptation. It estimates the hyperparameters of the model and simultaneously train the weights.

In the real world, the target domain could be divided into multiple subdomains based on time and scene, where each subdomain exhibits different statistical characteristics while sharing the same high-level semantics. In such cases, the above domain adaptation methods are unable to extract representative domain invariant features, resulting in poor generalization performance across multiple domains.

III. DOMAIN ADAPTATION DATASET

The diversity of the dataset is crucial for the performance of the motion deblurring methods. The mainstream motion

blur dataset, e.g., GoPro [16], HIDE [38] and REDS [17] are synthetic. To be specific, these datasets generate blurry images by accumulating sharp frames in exposure time, leading to blur kernels that differ from those observed in real-world scenarios. Thereby, this discrepancy between the synthetic and real-world blur kernels can lead to a degradation in deblurring performance when testing in natural scenes. It motivates the need for more diverse and representative datasets in the motion deblurring field.

TABLE I
MAINSTREAM MOTION BLUR DATASETS AND MD DATASET

	GoPro	HIDE	REDS	BS-RSCD	MD (Ours)
Year	2017	2019	2019	2021	2022
Num	3214	8422	30000	4000	10000
Size	1280×720	1280×720	1280×720	640×460	1920×1080
Syn/Real	Syn	Syn	Syn	Real	Real
Use DTS	✗	✗	✗	✗	✓

Rim *et al.* [10] and Zhong *et al.* [11] use beam-splitter acquisition system to collect real-world motion blur datasets. However, the process of capturing motion blur datasets with a beamsplitter is inherently complex, which requires precise calibration of beamsplitter parameters. Moreover, beamsplitters are heavy to move, making it difficult to collect diverse scenes. For example, the low-light environment, foggy weather and rainy weather are not considered in their datasets, which are indispensable in the real world.

To alleviate this dilemma, we collect a real-world motion blur dataset. It is further augmented via the domain transfer strategy to construct a multi-domain dataset (MD dataset). MD dataset is the first motion blur dataset using domain transfer strategy. Fig. 2 displays some images of the proposed MD dataset. The effect of the MD dataset is demonstrated in the ablation study part of Section V.

A. Data Collection

A simple collection device is beneficial for constructing diverse datasets. We utilize two HDR-CX900E cameras, one tripod, one ND filter, and one quick release plate to collect real-world motion blur data. Specifically, the quick release plate is attached to the tripod, allowing the two cameras to be mechanically aligned in parallel. The shutter speed of the two cameras is adjusted so that the other camera parameters are identical. The shutter time of the cameras used to capture blurry images and clear images is 40ms and 5ms.



Fig. 3. Domain transferred results (low-light domain). The first row represents the normal domain images and the second row represents the domain transferred results in the low-light domain. Meanwhile, the degree of illuminance improves as the column increases.

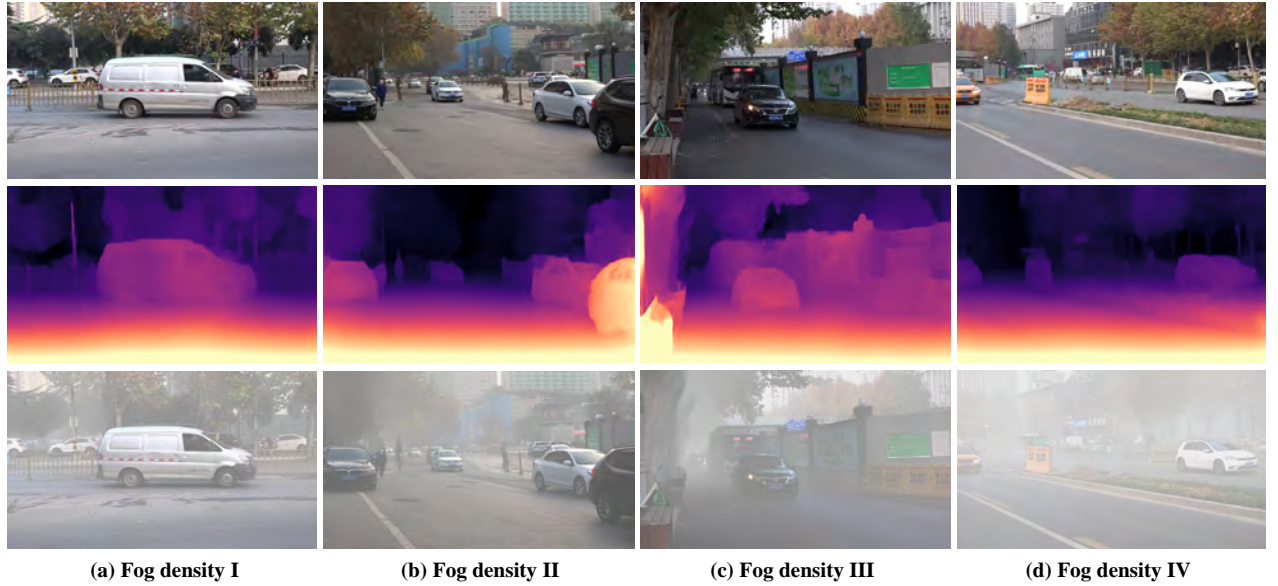


Fig. 4. Domain transferred results (foggy domain). The first row represents the normal domain images and the second row represents the corresponding depth map. The third row represents the domain transferred results in the foggy domain. Meanwhile, the foggy density improves as the column increases.

To ensure that the amount of the light getting into the two cameras is identical, an ND filter is installed in front of the camera with a longer exposure time to decrease its brightness. Since the cameras are positioned parallel to each other, image registration is performed to align the image pairs. Specifically, this is accomplished by using the SIFT [39] algorithm to detect features, followed by the KNN [40] algorithm to match them. Then, we compute the homography matrix of the image pair. Finally, the homography matrix is utilized to generate the aligned image pair. It can be observed in Table I that we collect a total of 5000 image pairs from various angles, focal lengths, and street scenes, with 1920×1080 resolution. Compared to mainstream motion blur datasets, except for REDS, the MD dataset contains more images and each image has larger image sizes.

B. Domain Adaptation

Although the collected dataset contains numerous scenarios, its diversity is not sufficient to cover all the scenes encountered

in the real world. Specifically, the collected dataset only contains normal illuminance and weather. While weather and illuminance changes do not impact the blur information of moving objects, they do have an influence on the ability of the model to capture the blur kernel. For instance, a moving object is more easily deblurred in a normal environment compared to low-light conditions. To increase the diversity of the dataset and maintain the deblurring capability in challenging conditions such as rain, fog, or low-light scenarios, the dataset is transferred to multiple domains.

Firstly, we leverage CoMoGan proposed by Pizzati *et al.* [41] to transfer the collected dataset from the source domain (normal domain) to the target domain (low-light domain). Instead of a point-to-point mapping ($X \rightarrow Y$, X and Y represent the source image and target image.), CoMoGan introduces the Functional Instance Normalization layer to disentangle image content from source image. This allows for the generation of continuous transferred images controlled by ϕ , namely $X \rightarrow Y(\phi)$. It guarantees both diversity and



Fig. 5. Domain transferred results (rainy domain). The first row represents the normal domain images. The second row represents the domain transferred results in the rainy domain. Meanwhile, the rainy level improves as the column increases.

smoothness of the dataset. It is formulated as

$$Y(\phi) = (1-\alpha)x + \alpha\Omega(x, \text{HSK}(\phi) + \text{corr}(\phi)) + \text{corr}(\phi), \quad (1)$$

where x represents the normal domain image. $\Omega(\cdot)$ represents the day-to-night tone mapping. $\text{HSK}(\cdot)$ represents the Hosek radiance model [42] and $\text{corr}(\cdot)$ represents the asymmetrical hue correction. Since other methods are not able to generate continuous low-light domain images, CoMoGan is chosen to transfer the collected dataset. Fig. 3 presents the visual results of the transferred dataset in the low-light domain.

In order to generate realistic foggy domain images, accurate depth maps are needed as inputs for the standard geometric model [43]. The standard geometric model is computed as

$$I(z) = O(z)t(z) + L(1 - t(z)), \quad (2)$$

where $I(z)$ is the generated foggy image at pixel z . $O(z)$ is the normal domain image. $L(\cdot)$ is the atmospheric light, and $t(z)$ determines the amount of scene radiance depending on the distance $\ell(z)$, i.e., the depth map. $t(z)$ is formulated as

$$t(z) = \exp(-\beta\ell(z)), \quad (3)$$

where parameter β controls the fog density. Most current monocular depth estimation networks utilize CNNs to extract feature information. However, CNNs have limitations as they do not consider the characteristics of geometric depth information and the distribution of depth maps. In contrast, GCNdepth [44] leverages graph convolution networks to handle irregular image regions and generate accurate depth maps. Therefore, we combine the depth map produced by GCNdepth with the standard geometric model to transfer images from the source domain (normal weather) to the target domain (foggy weather). Following the same procedure as for low-light domain transfer, images with different foggy densities are generated. Fig. 4 shows the visual results in the foggy domain.

Due to the fact that current rainy weather images are synthesized artificially, following their approach, we generate rainy domain images using Adobe After Effects software. The process is formulated as

$$R(z) = O(z) + \gamma P(z), \quad (4)$$

where $R(z)$ and $O(z)$ represent the rainy domain image and normal domain image. $P(z)$ represents the position and

intensity of raindrops, and γ represents the transparency of the raindrops. Continuous rainy domain images could be generated by controlling the $P(z)$. Fig. 5 shows the rainy domain images.

Notably, although the real-world motion blur dataset is transferred to multiple domains via a domain transfer strategy, the blur kernel is identical to the normal domain images as the difference between the normal domain and the transferred domains is the illuminance, foggy density, or rainy level. They may interfere with the algorithms to capture the blur kernel, thereby increasing the difficulty of deblurring, but they do not alter the inherent information of the blur kernel. From this perspective, although the multi-domain images are generated by a domain transfer strategy, they still belong to real-world blur.

IV. METHODS

The proposed MDADNet consists of the domain adaptation (DA) module and meta deblurring (MDB) module. Its architecture is shown in Fig. 6. The DA module is designed to disentangle the domain adaptation features from the global features at the feature level, allowing for the effective restoration of blurry images in multiple domains. The MDB module complements the primary task and also enables the model to update parameters during the testing stage, further enhancing the deblurring ability of the proposed algorithm.

A. Domain Invariant Feature Module

As mentioned before, we construct a large motion blur dataset and adopt the domain transfer scheme to augment the domains, which improves the diversity of the dataset. However, the performance of deblurring models tends to decrease when the domain of the test set varies. Therefore, many studies have focused on extracting domain-invariant features in the domain adaptation field, which can disentangle domain-invariant features from global features. Wu *et al.* [45] propose a progressive disentanglement strategy to extract domain invariant features. However, the domain invariant features may not be representative since they only maximize the gap between the domain invariant features and domain relevant features, ignoring their representativeness. In this paper, DA module is designed to

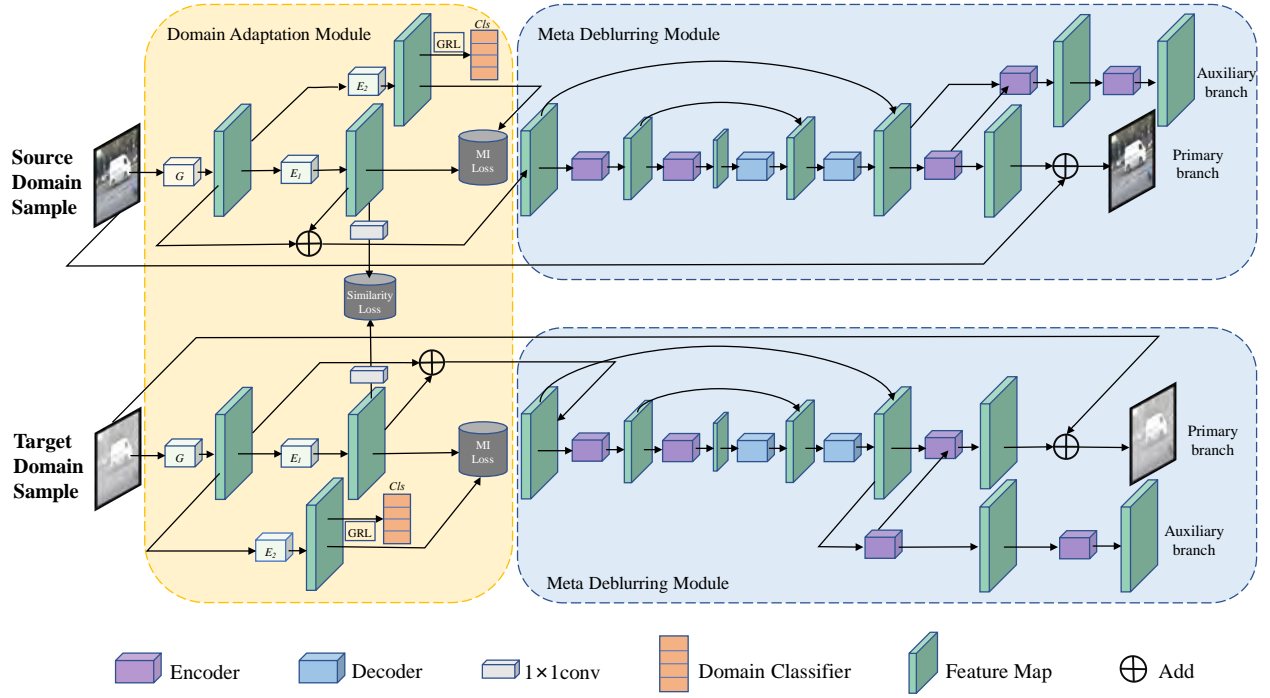


Fig. 6. Overview of the proposed MDADNet. It consists of the domain adaptation (DA) module and the meta deblurring (MDB) module. 'GRL' represents the gradient reverse layer. 'G' is the global feature extractor. 'E₁' and 'E₂' denote the domain invariant feature extractor and the domain relevant feature extractor. 'Cls' indicates the classifier. '⊕' represents the element-wise sum operation. DA module aims to extract representative domain invariant features to stabilize the performance of the MDADNet in multiple domains. MDB module aims to deploy the auxiliary branch to strengthen the deblurring ability of the primary branch.

extract much more representative domain invariant features, which enables the model to deblur well in multiple domains.

Architecture: Fig. 6 illustrates the architecture of the DA module in detail. It consists of three feature extractors, namely Global feature extractor G , domain invariant feature extractor E_1 , and domain relevant feature extractor E_2 , along with a domain classifier Cls . G contains three resblocks [46]. E_1 and E_2 contains one resblock [46]. Since domain relevant features $F_2 \in \mathbb{R}^{H \times W \times C}$ often include overall styles such as texture and structure, it requires a large receptive field to be extracted. Here, H, W , and C represent the height, width, and channel. Nevertheless, the pooling operation used to increment the size of the receptive field leads to a loss of spatial information and reduces the size of the feature map. Additionally, to combine the global feature $F_G \in \mathbb{R}^{H \times W \times C}$ with the domain invariant feature $F_1 \in \mathbb{R}^{H \times W \times C}$ as the output of the DA module, F_1 and F_G should have identical sizes. Considering the above factors, we adopt the dilated convolution layer [47] to replace the pooling operation in the downsampling stage. The domain classifier, which is employed to distinguish the category of the domain relevant feature, consists of three fully-connected layers.

Loss Function: Mutual information (MI) loss is a crucial loss function for measuring the correlation of two random variables. Here, MI loss is adopted to maximize the gap between the domain relevant features and the domain invariant features. It can be computed by Mutual Information Neural Estimation (MINE) approach [48]. It is described as

$$\mathcal{L}_{MI} = \frac{1}{n} \sum_{i=1}^n T_{\theta}(x^{(i)}, z^{(i)}) - \log \left(\frac{1}{n} \sum_{i=1}^n e^{T_{\theta}(x^{(i)}, z^{(i)})} \right), \quad (5)$$

where (x, z) is a pair of samples drawn from the joint distribution of domain invariant feature and domain relevant feature. n is the number of sample pairs. z represents the sample drawn from the marginal distribution of the domain relevant feature. T_{θ} is a neural network which is devised to compute MI loss.

Since the difference between the source domain image and target domain image lies in their illuminance or weather rather than the content, the domain invariant features of the source domain and target domain should be similar. To satisfy this condition, cosine loss is employed to measure the similarity of domain invariant features between the source domain and the target domain. It is defined as

$$\mathcal{L}_{sim} = 1 - \frac{1}{n} \sum_{i=1}^n \cos(x_s^{(i)}, x_t^{(i)}), \quad (6)$$

where x_s and x_t represent the domain invariant feature of the source domain and target domain, respectively.

As cross entropy loss is commonly used in the classification task, it is adopted as the loss function of the domain classifier. It is computed as

$$\mathcal{L}_{cls} = -\frac{1}{n} \sum_{i=1}^n (y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)), \quad (7)$$

where y and \hat{y} represent the label of the domain relevant feature and the output of the domain classifier.

B. Meta Deblurring Module

Meta auxiliary learning (MAXL) is beneficial for learning the adaptative weights in a meta learning manner. It is proposed by Liu *et al.* [49]. The authors claim that the interaction between the primary and auxiliary tasks can be seen as a type of meta-learning, thus it is called meta auxiliary learning. Chi *et al.* [26] adopt MAXL to improve the performance of the primary branch. However, their model is CNN-based, and they do not integrate the primary and auxiliary branches (i.e., the primary and auxiliary branches are separately updated during the training stage), which could result in an imbalance between the two tasks. In this work, we follow the MAXL theory and propose the meta deblurring module for motion deblurring. To be specific, to better extract context features, we introduce the Swin Block [50] in the downsampling stage. Furthermore, to combine the primary task and auxiliary task, different weights are assigned to these two branches.

Architecture: As shown in Fig. 6, the output of the DA module is used as the input of the MDB module to reduce domain distractions. Then, we utilize an encoder-decoder architecture as the backbone of the module. The encoder and decoder consist of Swin Block [50] and deconv layers [51], respectively. To effectively utilize low-level vision features during the upsampling stage, we employ skip connections to merge the feature map generated by the encoder with the corresponding feature map produced by the decoder. Inspired by previous MAXL work [26], we design two branches, the primary branch and auxiliary branch, to enhance the performance of motion deblurring. The primary branch and auxiliary branch consist of one and two Swin Blocks, respectively.

Meta Auxiliary Branches: To take advantage of MAXL, we design a MAXL framework for motion deblurring. As shown in Fig. 6, the primary branch and auxiliary branch share most of the parameters. It allows for the self-update of the primary branch when the auxiliary branch iterates. The output of the primary branch contains residual information, which is element-wise added to the blurry image as the restored image. Then, the residual information is transmitted to the auxiliary branch, whose task is to reconstruct the blurry image. Hence, the loss function of the primary task is defined as

$$\mathcal{L}_p = \frac{1}{n} \sum_{i=1}^n \|I_c^{(i)} - \hat{I}_c^{(i)}\|_1, \quad (8)$$

where I_c means the clear image, and \hat{I}_c denotes the deblurred image. The loss function of the auxiliary task is defined as

$$\mathcal{L}_a = \frac{1}{n} \sum_{i=1}^n \|I_b^{(i)} - \hat{I}_b^{(i)}\|_1, \quad (9)$$

where I_b is blurry image and \hat{I}_b means the reconstructed image. As the loss function only requires the blurry image, it enables the model to update itself during the testing stage. Different from Chi *et al.* [26], to train the MDADNet, we take both the residual image and the reconstructed image into consideration. The loss function of the MDB module is formulated as

$$\mathcal{L}_{MDB} = \delta \mathcal{L}_p + (1 - \delta) \mathcal{L}_a. \quad (10)$$

δ is penalty coefficient that balance the multiple tasks. Here, based on the experimental results recorded in Table II, it can be observed that setting the value of δ to 0.8 yields the best result.

TABLE II
QUANTITATIVE RESULTS OF DIFFERENT δ VALUES.

δ	0.2		0.4		0.6		0.7		0.8		0.9	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
MDADNet	27.86	0.861	28.74	0.881	29.38	0.895	29.83	0.902	30.21	0.916	30.04	0.909

The total loss function combines the MI loss, similarity loss, classified loss, and the MDB Module loss. It is defined as

$$\mathcal{L} = \mathcal{L}_{MDB} + \mathcal{L}_{MI} + \mathcal{L}_{sim} + \mathcal{L}_{cls}. \quad (11)$$

The training of the proposed MDADNet is implemented by minimizing \mathcal{L} .

V. EXPERIMENTS

In this section, we give the quantitative and qualitative results of the proposed MDADNet on four mainstream motion blur datasets and the constructed dataset (MD dataset). Meanwhile, the effects of the MD dataset and each component of MDADNet are displayed and analyzed in the ablation study.

A. Implementaion Details and Metrics

To compare with the other motion deblurring methods on the MD dataset, the proposed MDADNet and the other methods are trained on the MD dataset. The MD dataset has 10000 image pairs, including 5000 normal image pairs, 1700 foggy domain image pairs, 1700 low-light domain image pairs, and 1600 rainy domain image pairs. It is divided into 7000, 1500, and 1500 image pairs for the training set, validation set, and testing set, respectively. Then, following by previous methods [58], [56], we train the MDADNet on the GoPro dataset, and the GoPro-trained MDADNet is tested on public datasets.

Random rotation, random flip, and random crop are adopted as augmentation. As for the optimizer, we choose the SGD optimizer with momentum [59] 0.9. The initial learning rate is set to 0.0001, and it reduces 2×10^{-6} every 10 epochs. We run 1000 epochs to train the model. The resolution of all images is resized to 1280×720, and the batch size is set to 8. Note that the normal domain image and the corresponding transferred domain image are simultaneously fed into the network to better obtain representative domain adaptation features in a batch. Then, we set the number of self-update to 5 during the testing stage to obtain the best results. The experiments are performed on NVIDIA RTX 3090 using the PyTorch framework. As PSNR and SSIM are mainstream metrics to evaluate the quality of images for motion deblurring, they are adopted as evaluation metrics in this paper.

B. Effect of Different Domain Levels

To explore the impact of various domain levels on deblurring performance, we compare the experimental results of the original model (i.e., the proposed network without the

TABLE III
QUANTITATIVE RESULTS OF THE ORIGINAL MODEL AND MDADNet AT DIFFERENT FOGGY DOMAIN LEVELS.

Method	Normal domain		Fog level I		Fog level II		Fog level III		Fog level IV	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Orig	30.10	0.911	29.83	0.900	29.67	0.897	29.34	0.885	28.51	0.855
MDADNet	30.17	0.913	30.12	0.911	30.11	0.911	30.06	0.908	29.78	0.899

TABLE IV
QUANTITATIVE RESULTS OF THE DEBLURRING METHODS ON MAINSTREAM MOTION BLUR DATASETS. FOLLOWING PREVIOUS WORKS, THE PROPOSED MDADNet IS TRAINED ON THE GoPro DATASET AND DIRECTLY TESTED ON PUBLIC DATASETS. ON THE MD DATASET, THESE DEBLURRING METHODS ARE TRAINED AND TESTED ON IT. THE BEST RESULT IS BOLDDED.

Method	GoPro		REDS		HIDE		RealBlur-J		MD (ours)	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
NAFNet [28]	33.69	0.967	29.09	0.867	31.87	0.951	29.21	0.884	30.14	0.909
Restormer [27]	32.92	0.961	—	—	31.22	0.942	28.96	0.879	29.64	0.902
MIMO-UNet [52]	32.45	0.957	—	—	29.99	0.930	27.63	0.837	29.56	0.896
DeblurGAN [53]	28.70	0.858	—	—	24.51	0.871	27.97	0.834	24.32	0.827
DeblurGAN-v2 [54]	29.55	0.934	—	—	26.61	0.875	28.70	0.866	25.43	0.846
DBGAN [55]	31.10	0.942	—	—	28.94	0.915	—	—	27.25	0.869
Nah <i>et al.</i> [16]	29.08	0.914	—	—	25.73	0.874	27.87	0.827	24.11	0.823
MPR-Net [56]	32.66	0.959	28.79	0.911	30.96	0.939	28.70	0.873	29.59	0.897
HINet [57]	32.71	0.959	28.83	0.862	—	—	—	—	29.89	0.901
DeepRFT [15]	33.23	0.963	—	—	31.42	0.944	28.90	0.880	29.57	0.903
MDADNet (ours)	33.74	0.969	29.15	0.869	31.92	0.952	29.43	0.891	30.21	0.916



Fig. 7. Effects of different foggy domain levels. The first row represents the blurry image. The second row shows the results of the original model. The third row shows the results of the MDADNet. The fourth row shows the ground truth.

DA module and the auxiliary branch) and the MDADNet on different foggy-level images.

Fig. 7 displays the visual comparison results between the original model and our proposed method on the normal domain and four different domain levels. It can be observed that the restoration results of the original model are similar to those of the MDADNet in the normal domain. However, as the foggy level increases, the restoration ability of the original model significantly degrades. In contrast, MDADNet maintains a stable performance in the first three domain levels, owing to its DA module that enables the model to extract domain-invariant features with strong cross-domain capability. Additionally, the MDB module can further optimize the deblurring results during the testing stage, allowing the MDADNet to effectively restore blurry images even as the domain level increases.

Table IV shows the quantitative analysis results of the original model and the MDADNet on the normal domain and

four different domain levels. For the first three domain levels, the numerical results of the original model on the PSNR and SSIM metrics decrease as the domain level increases. In contrast, MDADNet shows no obvious change, further demonstrating the above conclusion. Note that our proposed method shows a certain degree of performance degradation on the fourth domain level. This can be attributed to the high foggy level, which significantly interferes with the blurry information of the object. As a result, it affects the ability of the MDADNet to extract representative features, leading to performance degradation.

C. Comparison with the State-of-the-art Method

To demonstrate the superiority of the proposed MDADNet, it is compared with current motion deblurring methods on the collected dataset (MD dataset) and four public datasets (GoPro [16], HIDE [38], REDS [17], Realblur-J [10]). The quantitative results are recorded in Table IV. Since Fig. 8 and Fig. 9 only show the local area of the restored images, to better show the performance of the MDADNet, the entire deblurring results are displayed in Fig. 10.

1) *Results on MD Dataset:* The MDADNet and the other compared methods are trained and tested on the MD dataset. As shown in Fig. 8, the first row exhibits the deblurring results in the normal domain. The second column, third column, and fourth column show the deblurring results in the low-light, foggy, and rainy domains, respectively. The MDADNet performs exceptionally well compared to other motion deblurring approaches, especially in local areas such as car license plates, car logos, and human faces. In these regions, the MDADNet restores the deblurred area to a much clearer image compared to the other methods. DeblurGAN fails to deblur the image. Moreover, the MDADNet maintains

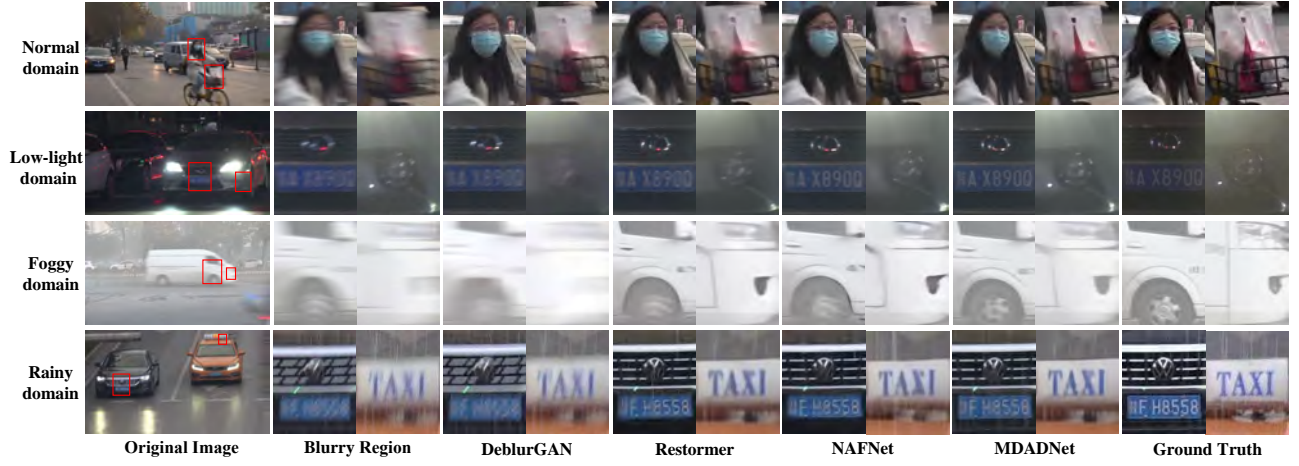


Fig. 8. Visual results on MD dataset. The results in the normal, low-light, foggy, and rainy domains are displayed from top to bottom rows. The second column represents the blurry region. The deblurring results restored by DeblurGAN, Restormer, NAFNet, and MDADNet (ours) are shown from the third column to the sixth column, respectively. The last column displays the clear images.

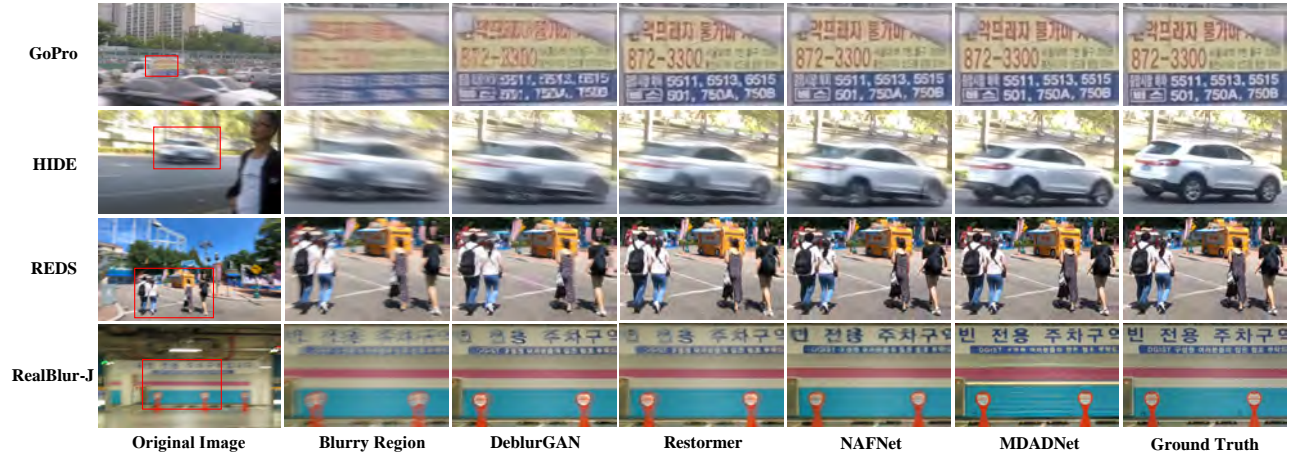


Fig. 9. Visual results on public datasets. The results on the GoPro, HIDE, REDS, and RealBlur-J datasets are displayed from top to bottom. The second column represents the blurry image. Columns 3-6 show the deblurring results restored by DeblurGAN, Restormer, NAFNet, and MDADNet (ours), respectively. The last column displays the clear images.

stable performance for motion deblurring in the low-light, foggy and rainy domains, whereas the other algorithms have obvious degradation compared with the normal domain. This is likely attributed to the cross-domain capability endowed to the model by the DA module, which allows the model to maintain stable performance in various domains. Additionally, during the testing stage, the MDB module enables the model to update the parameters, allowing it to restore more image details. The first column in Fig. 10 presents the entire restored image on the MD dataset, demonstrating that the MDADNet is capable of deblurring both local regions (e.g., car license) and large blurry objects (e.g., moving car) effectively. As Table IV describes, the MDADNet achieves the best results on both PSNR and SSIM metrics among the current methods, which further demonstrates the superiority of the MDADNet.

2) *Results on Public Datasets:* Following by previous methods, the MDADNet and the other compared methods are trained on the GoPro dataset and tested on public datasets. As Fig. 9 shows, each row exhibits the deblurring results



Fig. 10. Entire visual results on MD dataset and public datasets. The first row represents the blurry image. The second row shows the restored images. The last row is the corresponding clear images of the blurry images.

on four mainstream motion deblurring datasets. It is obvious that DeblurGAN performs poorly in restoring local areas. For instance, in Row 1 and Row 4, the text information of the images is reconstructed awful by DeblurGAN, Compared with it, NAFNet and Restormer are able to restore the text

information well. However, the outlines of the moving objects can not be disentangled from the background, causing the loss of the edge information in the deblurred images. Hopefully, the MDADNet could alleviate these problems. From Rows 2 and 3, it is obvious that the outlines of moving cars and persons are restored better than Restormer and NAFNet. Moreover, in Rows 1 and 4, the text information is effectively restored, and the deblurred images are closer to the ground truth than other methods, demonstrating that the MDADNet outperforms existing motion deblurring works. Apart from the local regions that Fig. 9 presents, Columns 2,3, and 4 in Fig. 10 also shows the entire deblurred results on the REDS, GoPro, and RealBlur-J datasets, which further demonstrates the outstanding performance of the MDADNet.

From Table IV, the MDADNet obtains the best quantitative results on three public datasets, namely GoPro [16], HIDE [38], and RealBlur-J [10]. On REDS dataset [17], it also achieves the best result on PSNR and slightly lower result on SSIM compared with MPR-Net. The remarkable deblurring performance of the MDADNet is likely due to the auxiliary branch in the MDB module. It enables the MDADNet to update itself to further restore the details. Meanwhile, the skip connection between the encoder and decoder can fuse the low-level and high-level features which contain contour and semantic features. This is beneficial for capturing the blur kernel of the blurry objects.

D. Ablation Study

In this section, ablation experiments are performed on the MD dataset, REDS dataset [17] and GoPro dataset [16] to further display the effectiveness of each component of the proposed method. Moreover, to intuitively show the efficiency that the MDADNet could handle various types of motion blur, global blurry and non-uniform blurry images are selected from the public datasets and MD dataset as the testing samples.

Table V and Fig. 11 provide the quantitative and qualitative results of each component of the proposed approach. To be specific, 1) Orig: the original dataset without using the domain transfer strategy and the proposed network without the DA module and the auxiliary branch; 2) DTS: the collected dataset uses the domain transfer strategy to augment the dataset; 3) DA: the domain adaptation module; 4) Aux: the MDB module adopts auxiliary branch. Note that the number of self-update is 5. As shown in Table V, each component of MDADNet improves the performance of deblurring, especially the DA module and the auxiliary branch in the MDB module. The DA module and the auxiliary branch achieve 1.43 dB and 1.25 dB improvements respectively. The visualizations from Fig. 11 further support this conclusion.

1) *Effect of the Collected Dataset Using Domain Transfer Strategy:* To demonstrate that the collected dataset using the domain transfer strategy is closer to the distribution of real-world blur than the other motion blur datasets, the MDADNet and the other deblurring methods are trained on the MD dataset, GoPro dataset, and REDS dataset, respectively. We follow the implementation details from Section V for training. To be fair, they are tested on the Real-blur dataset containing real-blur scenes.

TABLE V
ABLATION STUDIES OF THE MDADNet ON THE REDS, GoPro AND MD DATASET(OURS). THE BEST RESULT IS BOLDED.

Method	REDS		GoPro		MD (Ours)	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Orig	26.73	0.841	29.33	0.871	27.61	0.869
Orig+DTS	27.14	0.844	30.11	0.896	28.04	0.874
Orig+DA	27.81	0.853	31.25	0.923	28.89	0.887
Orig+DTS+DA	28.17	0.856	31.96	0.944	29.22	0.892
Orig+DTS+DA+Aux	29.15	0.869	33.74	0.969	30.21	0.916

Table VI reports the results of the MDADNet and the other deblurring methods trained on different datasets. It can be observed that the MD-trained methods achieve the best results both on PSNR and SSIM metrics. The quantitative results demonstrate the efficiency of the domain transfer strategy. Additionally, the MD dataset exhibits a closer resemblance to real-world blurs than other motion blur datasets, which is conducive to the motion deblurring task.

TABLE VI
QUANTITATIVE RESULTS OF THE MDADNet AND THE OTHER DEBLURRING METHODS ON THE REALBLUR-J TEST SET. THEY ARE TRAINED ON DIFFERENT DATASETS, NAMELY GoPro, REDS, AND MD DATASET, RESPECTIVELY. TO BE FAIR, THEY ARE TESTED ON THE REALBLUR-J TEST SET. THE BEST RESULT IS BOLDED.

Method	GoPro		REDS		MD (ours)	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
MDADNet	29.43	0.891	30.12	0.905	30.51	0.919
Restormer [27]	28.96	0.879	29.34	0.887	29.74	0.898
MIMO-UNet [52]	27.63	0.837	27.71	0.845	28.39	0.866
DeblurGAN [53]	27.97	0.834	28.15	0.852	28.57	0.871
DeblurGAN-v2 [54]	28.70	0.866	28.94	0.871	29.11	0.882
MPR-Net [56]	28.70	0.873	29.16	0.877	29.47	0.891
Nah et al. [16]	27.87	0.827	28.04	0.843	28.36	0.869
DeepRFT [15]	28.90	0.880	29.58	0.895	30.17	0.908

2) *Effect of the Domain Adaptation Module:* As mentioned in Section IV, the DA module could extract representative domain adaptation feature as the input of the MDB module. To better show the effect of the DA module, we visualize the feature map generated by the DA module and global feature extractor, respectively.

As shown in Fig. 12, the first row displays the input blurry images. The second and third rows show the heat maps of the global feature and the domain adaptation feature, respectively. From the visualization of the global feature, we observe that the global feature extractor has a high response to almost every region in the heat map. Compared with the global feature, only the region of blurry objects is activated by the domain invariant feature extractor in the domain adaptation feature. Notably, the input images belong to non-uniform blurry images. For non-uniform blurry images, only the moving object is blurry. It means that the blur kernel of the blurry object remains unchanged when the domain varies. Hence, the domain invariant feature of the non-uniform blurry images can be regarded as equivalent to the features of the blurry object. The visual results show that the DA module is able to extract

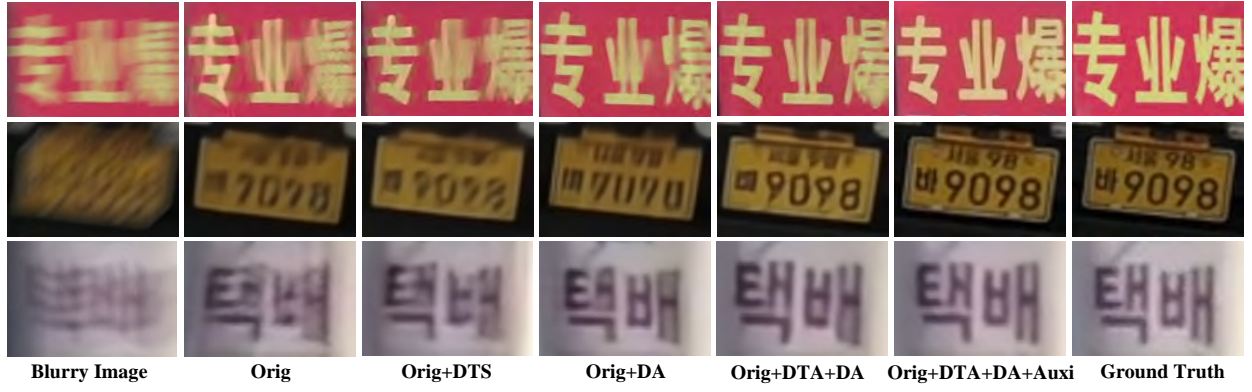


Fig. 11. Visual results of ablation studies on the REDS, GoPro and MD dataset. The first row represents the MD dataset and the second row denotes the REDS dataset. The last row is the GoPro dataset.



Fig. 12. Heat map of the domain adaptation feature. The first row is the blurry image. The second row and the third row are the heat map of the global feature and domain adaptation feature, respectively.

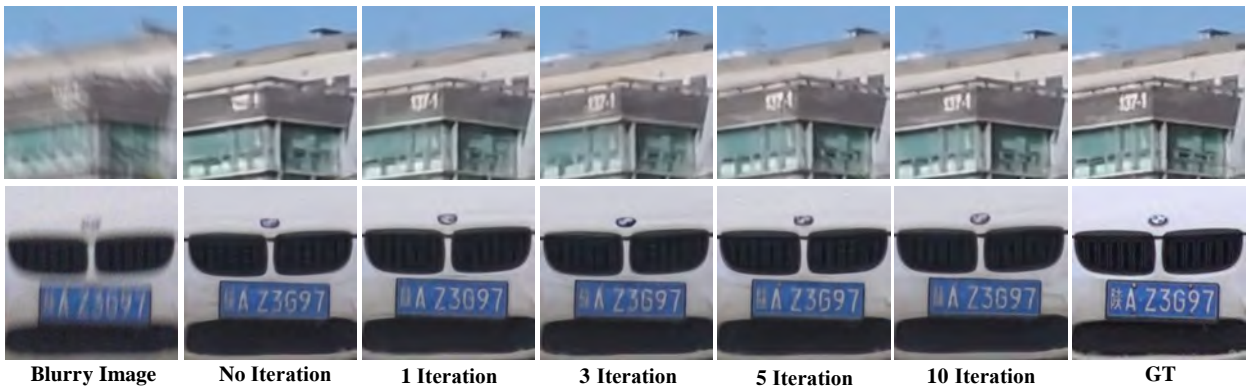


Fig. 13. Visual results of self-update during the testing stage. The first row shows the deburring results on the GoPro dataset. The second row represents the results on MD dataset (ours).

representative domain adaptation features, especially for non-uniform motion blur.

3) *Effect of the Meta Deblurring Module*: The MDB module improves the deblurring quality of the primary branch by leveraging the auxiliary branch. Additionally, the auxiliary

branch allows the MDADNet to quickly adapt to the distribution of the testing set, improving the deblurring ability. While Fig. 11 demonstrates the superiority of the MDB module, it does not explore the contribution of test adaptation. To further demonstrate its effectiveness, different numbers of self-updates

are visualized.

TABLE VII

QUANTITATIVE RESULTS OF DIFFERENT TIMES SELF-UPDATE ON THE MD DATASET AND GoPro DATASET. ITER IS THE ABBREVIATION FOR ITERATION

Dataset	No Iter		1 Iter		3 Iter		5 Iter		10 Iter	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Ours	29.75	0.902	29.72	0.900	30.03	0.910	30.21	0.916	30.22	0.916
GoPro	33.27	0.954	33.24	0.953	33.51	0.962	33.74	0.969	33.75	0.970

From Fig. 13 and Table VII, we observe that the quality of restored images improves after self-updates, except for images updated only once, possibly due to a domain gap between the testing and training sets. Besides, when the number of self-update exceeds 5, the performance nearly has no improvement both in visual and quantitative metrics. Fig. 13 contains both global and non-uniform blurry images, showing that test adaptation can effectively handle both types of motion blur. In summary, the self-update enables the model to quickly adapt the distribution of the testing set so that the performance for motion deblurring could be further improved.

E. Discussion

Although the MDADNet could stabilize the performance in multiple domains by extracting representative domain invariant features, its deblurring ability still suffers to some extent for extreme and unseen domains such as heavy noise and geometric distortion. One possible solution is to use a domain transfer scheme to augment the existing dataset with more domains. However, this approach may not cover all real-world domains. Another potential development is to build a large dynamic dictionary of blur kernels. With such a dictionary, the model could automatically capture the blur kernel and restore the blurry images, which may improve its deblurring ability.

VI. CONCLUSION

In this work, a real-world motion blur dataset is constructed. It is further augmented to low-light, foggy, and rainy domains by the domain transfer strategy, namely the multi-domain (MD) dataset. MD dataset is the first dataset containing multiple domains for motion deblurring. It empowers the MD-trained model to maintain stable performance in different weather and illuminance conditions. Besides, a Multi-Domain Adaptive Deblurring Network for motion deblurring is presented in this work. It consists of the domain adaptation (DA) module and the meta deblurring (MDB) module. The DA module plays a vital role in disentangling domain invariant features from global features, enabling the MDADNet to focus on the blur kernel and ignore the variations of the background domains. The MDB module leverages an auxiliary task to assist with the primary task of restoring blurry images. During the testing stage, it allows the model to quickly adapt to the distribution of testing images, which enhances the deblurring ability of the proposed method. Compared to previous works, MDADNet is more effective in handling interference from background domains and can further optimize images during the testing phase. This allows it to achieve superior deblurring

performance in natural scenes. Extensive experiments conclusively show that the MDADNet attains the best results for motion deblurring. In future work, we plan to transfer the motion blur dataset to more domains and focus on exploring more representative domain invariant features.

REFERENCES

- [1] C.-H. Liang, Y.-A. Chen, Y.-C. Liu, and W. H. Hsu, "Raw Image Deblurring," *IEEE Transactions on Multimedia*, vol. 24, pp. 61–72, 2020.
- [2] Y. Liu, F. Fang, T. Wang, J. Li, Y. Sheng, and G. Zhang, "Multi-Scale Grid Network for Image Deblurring with High-Frequency Guidance," *IEEE Transactions on Multimedia*, vol. 24, pp. 2890–2901, 2021.
- [3] J. Li, B. Yan, Q. Lin, A. Li, and C. Ma, "Motion Blur Removal with Quality Assessment Guidance," *IEEE Transactions on Multimedia*, vol. 23, pp. 2986–2997, 2021.
- [4] Y. Yuan, W. Su, and D. Ma, "Efficient Dynamic Scene Deblurring Using Spatially Variant Deconvolution Network with Optical Flow Guided Training," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3555–3564.
- [5] Q. Li, Y. Yuan, X. Jia, and Q. Wang, "Dual-Stage Approach Toward Hyperspectral Image Super-Resolution," *IEEE Transactions on Image Processing*, vol. 31, pp. 7252–7263, 2022.
- [6] Q. Li, M. Gong, Y. Yuan, and Q. Wang, "Symmetrical Feature Propagation Network for Hyperspectral Image Super-Resolution," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–12, 2022.
- [7] L. Xu, S. Zheng, and J. Jia, "Unnatural L0 Sparse Representation for Natural Image Deblurring," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 1107–1114.
- [8] X. Yu, F. Xu, S. Zhang, and L. Zhang, "Efficient Patch-Wise Non-Uniform Deblurring for a Single Image," *IEEE Transactions on Multimedia*, vol. 16, no. 6, pp. 1510–1524, 2014.
- [9] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman, "Removing Camera Shake from A Single Photograph," *ACM Transactions on Graphics*, vol. 25, pp. 787–794, 2006.
- [10] J. Rim, H. Lee, J. Won, and S. Cho, "Real-World Blur Dataset for Learning and Benchmarking Deblurring Algorithms," in *Proc. European Conference on Computer Vision*, 2020, pp. 184–201.
- [11] Z. Zhong, Y. Zheng, and I. Sato, "Towards Rolling Shutter Correction and Deblurring in Dynamic Scenes," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2021, pp. 9219–9228.
- [12] Y. Zhang and K. Hirakawa, "Blur Processing Using Double Discrete Wavelet Transform," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 1091–1098.
- [13] C. Min, G. Wen, B. Li, and F. Fan, "Blind Deblurring via A Novel Recursive Deep CNN Improved by Wavelet Transform," *IEEE Access*, vol. 6, pp. 69 242–69 252, 2018.
- [14] W. Zou, M. Jiang, Y. Zhang, L. Chen, Z. Lu, and Y. Wu, "SDWNet: A Straight Dilated Network with Wavelet Transformation for Image Deblurring," in *Proc. IEEE International Conference on Computer Vision*, 2021, pp. 1895–1904.
- [15] X. Mao, Y. Liu, W. Shen, Q. Li, and Y. Wang, "Deep Residual Fourier Transformation for Single Image Deblurring," *arXiv preprint arXiv:2111.11745*, 2021.
- [16] S. Nah, T. Hyun Kim, and K. Mu Lee, "Deep Multi-Scale Convolutional Neural Network for Dynamic Scene Deblurring," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3883–3891.
- [17] S. Nah, S. Son, S. Lee, R. Timofte, and K. M. Lee, "NTIRE 2021 Challenge on Image Deblurring," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2021, pp. 149–165.
- [18] S. Su, M. Delbracio, J. Wang, G. Sapiro, W. Heidrich, and O. Wang, "Deep Video Deblurring for Hand-Held Cameras," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1279–1288.
- [19] J. Sun, W. Cao, Z. Xu, and J. Ponce, "Learning a Convolutional Neural Network for Non-Uniform Motion Blur Removal," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 769–777.
- [20] 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, 2014.

- [21] O. Kupyn, V. Budzan, M. Mykhailych, D. Mishkin, and J. Matas, "DeblurGAN: Blind Motion Deblurring Using Conditional Adversarial Networks," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8183–8192.
- [22] D. Gong, J. Yang, L. Liu, Y. Zhang, I. Reid, C. Shen, A. van den Hengel, and Q. Shi, "From Motion Blur to Motion Flow: A Deep Learning Solution for Removing Heterogeneous Motion Blur," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2319–2328.
- [23] X. Zhang, F. Wang, H. Dong, and Y. Guo, "A Deep Encoder-Decoder Networks for Joint Deblurring and Super-Resolution," in *Proc. International Conference on Acoustics, Speech and Signal Processing*, 2018, pp. 1448–1452.
- [24] X. Tao, H. Gao, X. Shen, J. Wang, and J. Jia, "Scale-Recurrent Network for Deep Image Deblurring," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8174–8182.
- [25] P. Tran, A. T. Tran, Q. Phung, and M. Hoai, "Explore Image Deblurring via Encoded Blur Kernel Space," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2021, pp. 11956–11965.
- [26] Z. Chi, Y. Wang, Y. Yu, and J. Tang, "Test-Time Fast Adaptation for Dynamic Scene Deblurring via Meta-Auxiliary Learning," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2021, pp. 9137–9146.
- [27] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, and M.-H. Yang, "Restormer: Efficient Transformer for High-Resolution Image Restoration," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5728–5739.
- [28] L. Chen, X. Chu, X. Zhang, and J. Sun, "Simple Baselines for Image Restoration," *arXiv preprint arXiv:2204.04676*, 2022.
- [29] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks," in *Proc. IEEE International Conference on Computer Vision*, 2017, pp. 2223–2232.
- [30] S. Sankaranarayanan, Y. Balaji, C. D. Castillo, and R. Chellappa, "Generate to Adapt: Aligning Domains Using Generative Adversarial Networks," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8503–8512.
- [31] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Nets," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2014.
- [32] B. Sun, J. Feng, and K. Saenko, "Return of Frustratingly Easy Domain Adaptation," in *Proc. AAAI Conference on Artificial Intelligence*, 2016, pp. 2058–2065.
- [33] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-Adversarial Training of Neural Networks," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2096–2030, 2016.
- [34] X. Zhang, L. Zhou, R. Xu, P. Cui, Z. Shen, and H. Liu, "Towards Unsupervised Domain Generalization," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2022, pp. 4910–4920.
- [35] A. Kendall, Y. Gal, and R. Cipolla, "Multi-Task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7482–7491.
- [36] S. Mukherjee, R. Sarkar, E. Labryère, and J.-C. Olivo-Marin, "A Min-Max Based Hyperparameter Estimation for Domain-Adapted Segmentation of Amoeboid Cells," in *Proc. IEEE International Symposium on Biomedical Imaging*, 2021, pp. 1869–1872.
- [37] S. Mukherjee, R. Sarkar, M. Manich, E. Labryère, and J.-C. Olivo-Marin, "Domain Adapted Multi-Task Learning for Segmenting Amoeboid Cells in Microscopy," *IEEE Transactions on Medical Imaging*, 2022, DOI:10.1109/TMI.2022.3203022.
- [38] Z. Shen, W. Wang, X. Lu, J. Shen, H. Ling, T. Xu, and L. Shao, "Human-Aware Motion Deblurring," in *Proc. IEEE International Conference on Computer Vision*, 2019, pp. 5572–5581.
- [39] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, vol. 60, pp. 91–110, 2004.
- [40] T. Cover and P. Hart, "Nearest Neighbor Pattern Classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.
- [41] F. Pizzati, P. Cerri, and R. de Charette, "CoMoGAN: Continuous Model-Guided Image-to-Image Translation," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2021, pp. 14288–14298.
- [42] X. Huang and S. Belongie, "Arbitrary Style Transfer in Real-Time with Adaptive Instance Normalization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 1501–1510.
- [43] C. Sakaridis, D. Dai, and L. Van Gool, "Semantic Foggy Scene Understanding with Synthetic Data," *International Journal of Computer Vision*, vol. 126, no. 9, pp. 973–992, 2018.
- [44] A. Masoumian, H. A. Rashwan, S. Abdulwahab, J. Cristiano, and D. Puig, "GCNDepth: Self-Supervised Monocular Depth Estimation Based on Graph Convolutional Network," *arXiv preprint arXiv:2112.06782*, 2021.
- [45] A. Wu, Y. Han, L. Zhu, and Y. Yang, "Instance-Invariant Domain Adaptive Object Detection via Progressive Disentanglement," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 8, pp. 4178–4193, 2021.
- [46] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.
- [47] F. Yu and V. Koltun, "Multi-Scale Context Aggregation by Dilated Convolutions," *arXiv preprint arXiv:1511.07122*, 2015.
- [48] M. I. Belghazi, A. Baratin, S. Rajeshwar, S. Ozair, Y. Bengio, A. Courville, and D. Hjelm, "Mutual Information Neural Estimation," in *Proc. International Conference on Machine Learning*, 2018, pp. 531–540.
- [49] S. Liu, A. Davison, and E. Johns, "Self-Supervised Generalisation with Meta Auxiliary Learning," in *Proc. International Conference on Neural Information Processing Systems*, 2019, pp. 1677–1687.
- [50] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows," in *Proc. IEEE International Conference on Computer Vision*, 2021, pp. 10012–10022.
- [51] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus, "Deconvolutional Networks," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010, pp. 2528–2535.
- [52] S.-J. Cho, S.-W. Ji, J.-P. Hong, S.-W. Jung, and S.-J. Ko, "Rethinking Coarse-to-Fine Approach in Single Image Deblurring," in *Proc. IEEE International Conference on Computer Vision*, 2021, pp. 4641–4650.
- [53] O. Kupyn, V. Budzan, M. Mykhailych, D. Mishkin, and J. Matas, "Deblurgan: Blind Motion Deblurring Using Conditional Adversarial Networks," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8183–8192.
- [54] O. Kupyn, T. Martyniuk, J. Wu, and Z. Wang, "Deblurgan-v2: Deblurring (Orders-of-Magnitude) Faster and Better," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8878–8887.
- [55] K. Zhang, W. Luo, Y. Zhong, L. Ma, B. Stenger, W. Liu, and H. Li, "Deblurring by Realistic Blurring," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2737–2746.
- [56] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, M.-H. Yang, and L. Shao, "Multi-Stage Progressive Image Restoration," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2021, pp. 14821–14831.
- [57] L. Chen, X. Lu, J. Zhang, X. Chu, and C. Chen, "HINet: Half Instance Normalization Network for Image Restoration," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 182–192.
- [58] 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*, 2019, pp. 5978–5986.
- [59] S. Ruder, "An Overview of Gradient Descent Optimization Algorithms," *arXiv preprint arXiv:1609.04747*, 2016.



Kai Zhuang received the B.E. degree in computer science and technology and the M.E. degree in computer technology from Yanshan University, Qinhuangdao, China and Northwestern Polytechnical University, Xi'an, China, in 2020 and 2023 respectively. He is currently working toward the Ph.D. degree in the School of Artificial Intelligence, Optics and ElectroNics (iOPEN), Northwestern Polytechnical University, Xi'an, China. His research interests include computer vision, and machine learning.



Qiang Li (Member, IEEE) received the Ph.D. degree in computer science and technology from Northwestern Polytechnical University, Xi'an, China in 2022. He is currently a postdoc with the School of Electronic Engineering, Xidian University, Xi'an. His research interests include remote sensing image processing and computer vision.



Yuan Yuan (M'05-SM'09) is currently a Full Professor with the School of Artificial Intelligence, Optics and ElectroNics (iOPEN), Northwestern Polytechnical University, Xi'an, China. She has authored or co-authored over 150 papers, including about 100 in reputable journals, such as the IEEE TRANSACTIONS AND PATTERN RECOGNITION, as well as the conference papers in CVPR and BMVC. Her current research interests include visual information processing and image/video content analysis.



Qi Wang (M'15-SM'15) received the B.E. degree in automation and the Ph.D. degree in pattern recognition and intelligent systems from the University of Science and Technology of China, Hefei, China, in 2005 and 2010, respectively. He is currently a Professor with the School of Artificial Intelligence, Optics and ElectroNics (iOPEN), Northwestern Polytechnical University, Xi'an, China. His research interests include computer vision, pattern recognition and remote sensing.