# Variant-Depth Neural Networks for Deblurring Traffic Images in Intelligent Transportation Systems

Qian Wang, *Student Member, IEEE*, Cai Guo, *Student Member, IEEE*, Hong-Ning Dai , *Senior Member, IEEE*, and Min Xia , *Senior Member, IEEE*

***Abstract*— Intelligent transportation systems (ITS) with surveillance cameras capture traffic images or videos. However, images or videos in ITS often encounter blurs due to various reasons. Considering resource limitations, although recent technologies make progress in image-deblurring, there are still challenges in applying image-deblurring models in practical transportation systems: the model size and the running time. This work proposes an artful variant-depth network (VDN) to address the challenges. We design variant-depth sub-networks in a *coarse-to-fine* manner to improve the deblurring effect. We also adopt a new connection namely *stack connection* to connect all sub-networks to reduce the running time and model size while maintaining high deblurring quality. We evaluate the proposed VDN with the state-of-the-art (SOTA) methods on several typical datasets. Results on Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM) show that the VDN outperforms SOTA image-deblurring methods. Furthermore, the VDN also has the shortest running time and the smallest model size.**

***Index Terms*— Intelligent transportation systems (ITS), traffic image processing, image deblurring, variant-depth neural networks.**

## I. Introduction

THERE are growing interests in intelligent transportation systems (ITS), which play an important role in fostering smart cities and industrial systems [1], [2], [3]. Meanwhile, the recent advances in the Internet of Things, surveillance cameras, artificial i ntelligence, a nd 5 G c ommunications have also promoted the development of ITS and connected vehicles [4], [5]. Take surveillance cameras as an example. Various surveillance cameras, e.g., traffic e nforcement cameras, bayonet cameras, skynet monitoring cameras, have been widely applied in many scenarios in ITS, such as recognizing license plates of vehicles, tracing the trajectory of vehicles, identifying traffic signs, monitoring lanes, and detecting vehicles [6], [7], [8]. After analyzing images and videos collected by cameras via computer vision and deep learning algorithms, traffic characteristics (e.g., density, trajectory, and the speed of vehicles) can be extracted so that intelligent decisions can be made at vehicles or at ITS.

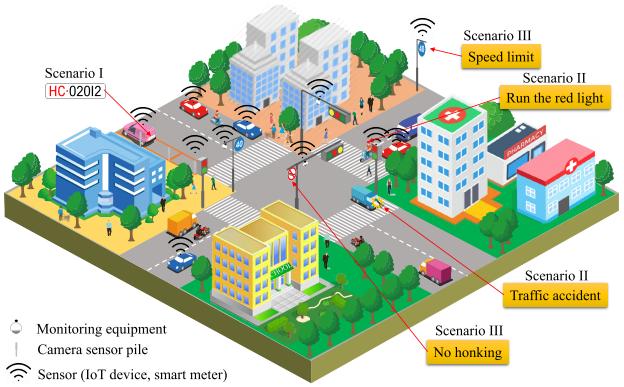


Fig. 1. Image-deblurring scenarios in intelligent transportation systems: (I) license-plate recognition, (II) traffic-accident identification, (III) traffic-sign recognition.

However, images and videos taken by the cameras often encounter blurs due to complex conditions such as vehicles moving, shaking cameras, or adverse weather conditions (e.g., fog, rain, and snow) [9], [10]. Blurry images are harmful to the development of ITS and autonomous vehicles. Take Fig. 1 as an example. In Scenario I, it is necessary to recognize the license plate of a vehicle in a parking management system while the blurry image of a license plate (due to the movement of a vehicle) often causes difficulty in obtaining accurate information about the vehicle. Scenario II shows that it is important to capture and identify images of vehicles or pedestrians in traffic accidents. However, the images taken by the camera are also blurry owing to multiple complex factors such as moving objects and shaking cameras. Scenario III depicts that autonomous vehicles need to recognize traffic signs while the blurry images often lead to challenges in traffic-sign recognition especially when the vehicle is moving at a high speed.

There are a line of researches on image deblurring. For example, conventional image-deblurring methods such as blind deblurring and non-blind deblurring approaches demonstrate excellent performance [11], [12], [13] while they often suffer

This work was supported in part by the Science and Technology Planning Project of Guangdong Province of China under Grant 2022A1515011551; in part by the Natural Science Foundation of Guangdong Province of China under Grant 2021A1515011091; in part by the Project of Educational Commission of Guangdong Province of China under Grant 2020ZDZX3056, Grant 2021KTSCX07, and Grant 2021KQNCX051; and in part by the Doctor Starting Fund of Hanshan Formal University, China, under Grant QD20190628. The Associate Editor for this article was S. H. A. Shah. *(Qian Wang and Cai Guo contributed equally to this work.) (Corresponding author: Hong-Ning Dai.)*

Qian Wang is with the Faculty of Innovation Engineering, Macau University of Science and Technology, Taipa, Macau (e-mail: anrogim@outlook.com).

Cai Guo is with the School of Computing and Information Engineering, Hanshan Normal University, Chaozhou 521000, China (e-mail: c.guo@hstc.edu.cn).

Hong-Ning Dai is with the Department of Computer Science, Hong Kong Baptist University, Hong Kong (e-mail: hndai@ieee.org).

Min Xia is with the Department of Engineering, Lancaster University, LA1 4YW Lancaster, U.K. (e-mail: m.xia3@lancaster.ac.uk).

from huge time consumption. As a result, they may not be feasible in ITS which has high real-time requirements. Although recent advances of convolutional neural networks (CNNs) utilized for image-deblurring [14], [15], [16], [17], [18], [19], [20] also show the superior performance while most of them still require a substantial running time and a large model size. Most of the SOTA image-deblurring models suffer from bulky model size and high running time.

In this paper, we propose a new variant-depth sub-network (namely VDN) for image deblurring in ITS. The proposed VDN model can well address the above challenges owning to the following characteristics.

### A. Variant Depth

The VDN leverages several variant-depth sub-networks to achieve the *coarse-to-fine* deblurring effect. Particularly, the VDN uses the varied convolution kernels and different numbers of Residual Blocks (ResBlocks) in these sub-networks to process different-level deblurring information. In other words, the shallow sub-network processes the coarse deblurring features and the deep sub-network processes the fine-grained deblurring features. Therefore, these different depth sub-networks are concatenated together to make full use of the deblurring information. This design of variant-depth sub-networks can improve the image-deblurring effect.

### B. Stack Connection

We name the concatenation among all sub-networks as *Stack Connection*. Inspired by the dense connection in the work of [21], we design this connection that connects each sub-network to every other sub-network in a feed-forward fashion. In order to simplify the whole network, we set the channel number of the output of the input layer of each sub-network uniformly. Since each sub-network has a direct connection, the vanishing-gradient problem is alleviated, and the deblurring information propagation is enhanced and reused. In particular, *Stack connection* connecting the sub-network exploits deblurring information from every sub-network to improve the quality of deblurred images.

### C. No Image Pre-Processing Procedure Required

Since the VDN feeds in the original blurry image at each sub-network, the network has no need to slice an input image into multiple patches or transform the image into multi-scale inputs. Consequently, the VDN model achieves the outstanding coarse-to-fine image-deblurring effect. No image pre-processing procedure makes the VDN succinct and suitable for ITS applications.

We highlight the major contributions as follows.

- We present a new *coarse-to-fine* image-deblurring model composed of variant-depth sub-networks to accomplish deblurring traffic images. This new variant-depth network makes the deblurring model more compact and more effective than SOTA methods. The experiment results indicate the proposed *coarse-to-fine* deblurring model outperforms SOTA methods in terms of deblurring effect, running time, and model size.
- We design *stack connections* to connect sub-networks so as to effectively reduce the running time and model size by reusing information flows across different sub-networks. Both the small model size and short running time benefit ITS applications.
- The overall architecture is concise and effective due to the artful design and no image pre-processing procedure. This advantage makes the VDN feasible in ITS applications.
- The experiments demonstrate that the VDN model outperforms SOTA methods in terms of PSNR (31.17) and SSIM (0.9453) with the smallest model size and the shortest running time.

We organize the rest of the paper as follows. We briefly introduce related work in Section II. We then explain the detailed method of the proposed VDN model for image deblurring in ITS in Section III. Experimental results are shown in Section IV to demonstrate the effectiveness and advantages of the proposed approach. In the end, a conclusion on the proposed method and a discussion about the possible future directions are presented in Section VI.

## II. Related Work

This section briefly surveys the related studies on image/video surveillance in ITS and image-deblurring approaches.

### A. Video-Surveillance in Its

Video-surveillance systems have been widely used in urban ITS. Diverse video cameras are deployed at transportation infrastructures to obtain images or videos for further analysis [22]. The images and videos obtained in intelligent surveillance systems can be used for analyzing pedestrian behaviors [23], [24], [25], vehicle-trajectory prediction [26], the road safety [27], real-time traffic surveillance [28], urban traffic congestion [29] and reasoning as well as decision-making [30]. Moreover, the analysis of images and videos is also important in parking management systems [31], [32]. It is critical to obtain the sharp images or videos for lane or road recognition for autonomous vehicles and parking management systems. However, the transmitted images or videos often encounter blurry owing to many reasons, e.g., bad weathers (fog, rainy, and snowy) and the moving objects [10], [33].

### B. Conventional Image Deblurring

There are a number of studies on deblurring images. Conventional image-deblurring methods can be categorized into blind deblurring and non-blind deblurring approaches. Blind-deblurring methods that are often based on the unknown blur kernels try to estimate sharp latent images and blur kernels. Non-blind deblurring methods are based on the spatial-invariance deblurring kernel. The authors in [12] propose a unified probabilistic model of both blind and non-blind deconvolution to separate the errors that arise during image-noise estimation and blur-kernel

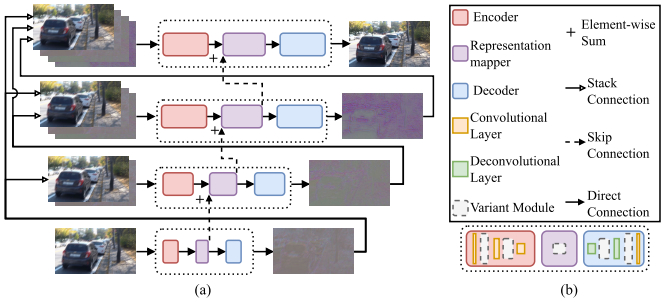


Fig. 2. The VDN consists of four variant-depth sub-networks in Fig. 2(a). The dotted frame denotes the sub-network. The sub-networks are built by different depths of encoders and decoders. The VDN connects all the sub-networks by Stack Connection. Two adjacent sub-networks are connected by concatenations noted by a solid line with a hollow arrow. At the same time, outputs of the shallower sub-networks are also fed into the deeper sub-networks. Fig. 2(b) depicts the structure of the encoder and the decoder.

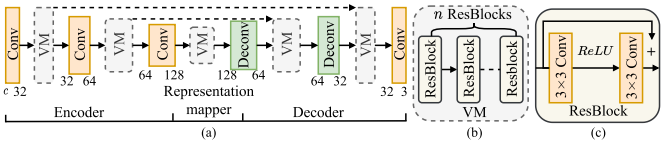


Fig. 3. Each sub-network of VDN consists of an encoder and a decoder, as shown in Fig. 3(a). In the encoder, there are 3 convolution layers (Conv.) and multiple variant modules. As shown in Fig. 3(b), each variant model consists of $n$ ResBlocks, where $n$ can be adjustable to fulfill different levels of feature processing. Fig. 3(c) shows the internal structure of a ResBlock. The decoder is a sandwiched structure consisting of two deconvolution layers (Deconv.), two variant modules and a convolution layer.

estimation. As indicated in [11], the blur kernels can be recovered by using transparency maps to get cues for object motion and performing blind-deconvolution with a prior on the alpha matte. But these methods cannot avoid using complex parameters. Considering the camera rotation-motion during exposure, the authors in [13] present a parameterized geometric model of the blurring process. They explain the spatially-varying blur according to the motion of 3-Dimensional (3D) rotational camera. Another similar framework is proposed in [34], in which the authors present a Motion Density Function for single image deblurring to estimate spatially non-uniform blur caused by camera shake. Despite the excellent performance on deblurring, their model is pretty time-consuming.

### C. CNN-Based Image Deblurring

Recently, the research interests in utilizing deep CNNs to the image or video deblurring issues are increasing, where "deep" means multiple CNN layers [35], [36]. It is shown in [14], [15], [16], [17], [18], [19], [37], and [20] that deep CNNs can achieve superior performance in image deblurring than conventional methods. One of recent important breakthroughs in deep networks is the deep residual networks proposed in [37]. Many evolved models based on deep residual networks have been widely devised and applied in many fields of computer vision and image processing, including object detection, image segmentation, image deblurring, and single-image super-resolution. The work of [38] stacks several residual units in their network as the feature mapping to achieve enhanced reconstruction. With respect to image deblurring, several recent studies devise image-deblurring methods based on deep residual networks. In particular, the authors in [15] use a small end-to-end regression block to build a deep network. Their model consisting of an auto-encoder and a generative network can remove the space-invariant and space-variant blur caused by camera motion. Meanwhile, a deep multi-scale CNN is devised in [16] to remove non-uniform blur for the realistic blurry images. In their model, a multi-scale loss function is designed to mimic the convolutional *coarse-to-fine* method to enhance convergence. Moreover, a simplified building block (namely ResBlock) is devised to boost the convergence speed at training time. However, there are too many ResBlocks stacked between two convolution layers, thereby leading to a quite deep network, through which some important features are lost.

### D. RNN-Based Image Deblurring

Besides deep CNNs, Recurrent Neural Networks (RNNs) also show their merits in sequential-information processing. Recently, RNNs are becoming an effective tool for image deblurring [17], [39]. In [17], the authors simplify the network structure presented in prior work [16] and propose a Scale-Recurrent Network (SRN) which can reduce training parameters. SRN can reduce the training difficulty and improve network stability through sharing network weights across scales in the network. Moreover, Convolutional Long Short-Term Memory cells are used to aggregate feature maps from *coarse-to-fine* scales. Later, the authors in [39] use RNNs for video deblurring. They propose to improve the accuracy of recurrent models by adapting the hidden states transferred from past frames to the current frame being processed.

### E. GAN-Based Image Deblurring

Recently, Generative Adversarial Networks (GANs) which show their advantages in preserving texture details and generating photo-realistic images [40], [41] have also been employed in image deblurring or dehazing [42], [43], [44]. In particular, the work [42] presents DeblurGAN based on GAN to image deblurring via restoring perceptually pleasing and sharp images, from both synthetic and real-world blurry images. On the basis of DeblurGAN, the authors in [43] devise an improved version of motion deblurring. GANs have also been utilized in image dehazing proposed by authors in [44]. In their work, the generator network is designed in dense connection combined with fine-scale and coarse-scale information.

Although existing approaches have made great progress in image deblurring, there still exist several challenges: 1) input images have to be transformed to achieve different levels of deblurring effects ascribed to the same sub-networks in their models, which usually leads to unsatisfactory deblurring effects for some kind blurry images; 2) expensive running times are required for the state-of-art deblurring models; 3) the model sizes are bulky due to complex models. All the above three challenges lead to the difficulty of widely deploying image/video surveillance systems in ITS.

## III. Image Deblurring Model for ITS

This section presents the technical details about the VDN model which can be applied in ITS. Section III-A first overviews the VDN, and Section III-B then presents the details of the VDN. Section III-C next introduces the loss functions and Section III-D investigates the variant-depth effects.

### A. Overview of Variant-Depth Method

We present the proposed VDN for image deblurring applied in ITS. Inspired by the *coarse-to-fine* concept, we design the VND with a multi-level network to achieve the *coarse-to-fine* deblurring effect. In particular, the VDN model is built with four sub-networks, each becoming deeper from the first (the shallowest) sub-network to the fourth (the deepest) one. In this artful design, the shallow sub-network processes coarse features while the deep sub-network processes fine-grained features. Different from multi-scale networks [16], scale-recurrent networks [17], and hierarchical multi-patch networks [45], the proposed network has no pre-processing procedure, e.g., multi-scale transformation or multi-patch fragmenting for the original inputs. As a result, the proposed model can preserve main features of the original input images at each sub-network. To improve the effectiveness of the *coarse-to-fine* method, the VDN artfully uses a *Stack Connection* to connect sub-networks. This connection allows a deeper sub-network continuously utilize the processed information from a shallower sub-network. In other words, the deblurring information flows are reused effectively. Furthermore, this connection will not significantly increase the number of parameters and running time. Thus, the VDN not only has a small model size and short computing time, but also outputs high-quality deblurring images. These advantages make the VDN be suitable for ITS.

In summary, there are three important novelties in the proposed VDN different from recent studies. First, the VDN is simpler than others owing to the depth of the whole network being shallower than others. Meanwhile, the VDN has no requirement for image-pre-processing (unlike multi-scale and multi-patch methods, which require partitioning images into patches or downscaling images). In other words, the VDN uses a simple and effective framework to address a complicated motion deblurring task. Second, the VDN has variant modules in each sub-network to control the depth of each sub-network so as to progressively enhance the deblurring effect with the increased depth of the sub-networks. Third, the VDN connects all its sub-network with stack connections. The stack connection can make the deeper sub-networks reuse the deblurring information of the shallower sub-networks so as to improve the deblurring effect progressively. As a result, the deblurred image is sharply close to the ground truth.

Fig. 2(a) depicts the concise architecture of the proposed ITS image deblurring model, VDN. It consists of the encoder-and-decoder structure in each sub-network (i.e., the red and blue blocks denote the encoder and the decoder, respectively) sandwiching a representation mapper (i.e., the purple block). The depth of each encoder and each decoder is different in each sub-network. Therefore, the depths of all sub-networks are becoming deeper from the first sub-network (i.e., the shallowest) to the fourth one (i.e., the deepest). Fig. 2(b) depicts the detailed structure of the sub-network. The encoder contains three convolutional layers staggering two variant modules. The representation mapper is a variant module. The decoder consists of two pairs of staggered deconvolutional layers and variant modules followed by a convolutional layer. The variant module (explained in Section III-B.1) consists of ResBlocks. The number of ResBlocks in each variant module is adjustable, thereby the depth of each sub-network controllable. The VDN connects all the sub-networks by the Stack Connection. Two adjacent sub-networks are connected by concatenations denoted as the solid line with a hollow arrow in Fig. 2(a). At the same time, outputs of the shallower sub-networks are also fed into the deeper sub-networks. Thus, the deblurring information of each sub-network is effectively reused without significantly increasing parameters. The deepest sub-network gains enough deblurring information to ensure the output vehicle image is quite close to the sharp vehicle image.

### B. VDN Details

*1) Encoder and Decoder:* We build the sub-network with an encoder-decoder structure indicated in Fig. 3(a). The encoder contains three pairs of staggered convolutional layers and variant modules. The decoder consists of a convolutional layer and two pairs of staggered deconvolutional layers with variant modules. The details of a variant module are exhibited in Fig. 3(b). As a core component of VDN, each variant module contains $n$ ResBlocks. And $n$ is adjustable to achieve different levels of processing the deblurring features. When the value of $n$ is increasing, the depth of the sub-network is growing. The shallow sub-network processes the coarse features and the deep sub-network processes the fine features. In other words, the processed deblurring features are delivered from the shallowest sub-network to the deepest sub-network in an accumulative form to achieve *coarse-to-fine* deblurring effect. In this *coarse-to-fine* manner, the VDN gets the restored sharp vehicle or traffic images. Fig. 3(c) expresses the detailed structure of the ResBlock used in the proposed variant module.

In a variant module consisting of $n$ ResBlocks, let $x_m$ and $y_m$ denote the input and output of the $m$-th ResBlock, respectively (where $m = 1, \ldots, n$). Let $w_m^1$ and $b_m^1$ denote the weight and the bias of the first layer in the $m$-th ResBlock, respectively. For the second layer in the $m$-th ResBlock, let $w_m^2$ and $b_m^2$ denote its weight and bias, respectively. Thus, the output of the $m$-th ResBlock is shown as follow,

$$y_m = x_m + w_m^2 \mathcal{G}(w_m^1 x_m + b_m^1) + b_m^2, \tag{1}$$

where $\mathcal{G}$ is an activation function (we choose the rectified linear unit (ReLU) as the activation function).

The input of the $(m+1)$-th ResBlock is the output of the $m$-th ResBlock, i.e., $x_{m+1} = y_m$. Deriving this principle recursively, the relationship between the $m$-th ResBlock's output and the $l$-th ResBlock's input is expressed as

$$x_{m+1} = x_l + \sum_{i=l}^{m} (w_i^2 \mathcal{G}(w_i^1 x_i + b_i^1) + b_i^2). \tag{2}$$

We then derive the difference value (D-value) between the input and output of $n$ ResBlocks. Let $\mathcal{R}_m$ denote the residual function of the $m$-th ResBlock. In particular, the D-value is equal to the sum of outputs of all previous residual function. Then the D-value from the input (the first ResBlock) to the output (the last ResBlock) is written as follows,

$$y_n - x_1 = \sum_{i=1}^{n} \mathcal{R}_i(x_i). \tag{3}$$

The higher value of $n$ increases the order of residual mapping function $\sum_{i=1}^{n} \mathcal{R}_i(x_i)$. The higher-order function with a more complex representation capability is easier to optimize.

The proposed VDN model uses the variant module to accomplish the *coarse-to-fine* deblurring effect. In the same sub-network of the VDN, $n$ of the variant module has the same value to achieve the same-level nonlinear transformation. The increased depths of the sub-networks increase $n$.

*2) Stack Connection:* Each depth-variant sub-network is stack-connected with another sub-network. Such connection is named as *Stack Connection*. Each sub-network is essentially a variant of the basic model of VDN as described in Section III-B.1 with the output fed into another sub-network. Therefore, in the delicate design, the deepest sub-network (i.e., the fourth sub-network) takes all other sub-networks outputs as its input. And each deeper sub-network takes all shallower sub-networks outputs as their inputs. In addition to unifying the channel numbers of the output of the input layer of each sub-network, we also added an intermediate skip connection between the sub-networks. *Stack Connections* are denoted by concatenation operations, each of which is the solid line with a hollow arrow pointing to the next sub-network as shown in Fig. 2.

The derivation of the basic *Stack Connection*. Let $\mathsf{Net}_{E_i}$, $\mathsf{Net}_{D_i}$, and $\mathsf{Net}_{R_i}$ denote the encoder, the decoder, and the representation mapper at $i$-th sub-network, respectively, and $B$ denote the input blurry image for the encoder. For the first sub-network (the bottom row in Fig. 2), the equation is written as:

$$\begin{aligned} E_i &= \mathsf{Net}_{E\_i}(B) \\ r_i &= \mathsf{Net}_{R\_i}(E_i), \quad \text{when } i = 1, \\ D_i &= \mathsf{Net}_{D\_i}(r_i) \end{aligned} \tag{4}$$

where the output of the encoder $E_i$ is used as the input of $\mathsf{Net}_{R_i}$, $r_i$ is the output of the representation mapper, and $D_i$ is the output of the decoder in the first sub-network.

For next sub-networks (from the second to the fourth sub-networks), the input of the representation mapper is the output of $\mathsf{Net}_{R_i}$ of the last sub-network with the addition with $\mathsf{Net}_{E_i}$ of the current sub-network (in Fig. 2). This basic *Stack Connection* can be extended to a general one with $i > 1$.

$$\begin{aligned} E_i &= \mathsf{Net}_{E\_i}\big(\mathsf{cat}[B, D_1, \ldots, D_{i-1}]\big) \\ r_i &= \mathsf{Net}_{R\_i}(E_i + r_{i-1}), \quad \text{when } i > 1, \\ D_i &= \mathsf{Net}_{D\_i}(r_i) \end{aligned} \tag{5}$$

where $\mathsf{cat}[\cdot]$ denotes the concatenation operation. Thus, the input of the $i$-th sub-network (i.e., the argument of $E_i$) is essentially the concatenation of blurry image $B$ with all the outputs from other shallower sub-networks. At the same time, the output of the $(i-1)$-th representation mapper and the output of the $i$-th encoder are added as the input of the $i$-th representation mapper.

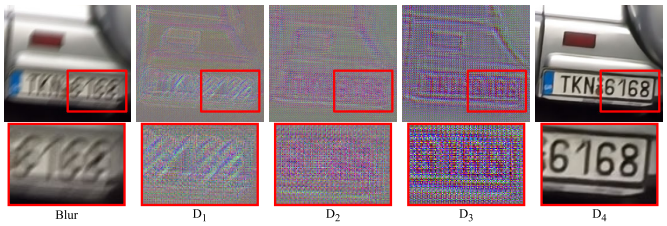


Fig. 4. Variant Depth Effect. The top-row images are the outputs of the four sub-networks of the VDN model. The first image on the leftmost is the blurry image, and $D_i$ indicates the outputs of each sub-network of VDN. The bottom-row pictures (in red boxes) are the magnified views of images. The output images become sharper, less sparse, and richer in color after the level-by-level processing.

As shown in Fig. 2(a), the *stack connection* denoted as the hollow-arrow line can be illustrated in an intuitive manner as follows. The output of the first sub-network is "inserted" into the second sub-network, and the outputs of both the first and the second sub-networks are "inserted" into the third sub-network, and so on. These four sub-networks evolve from the basic structure with different values $n$ of the variant module at each sub-network. Among the four sub-networks, the *coarse-to-fine* information from the shallower sub-network is delivered to all the rest deeper sub-network. The features learned by each shallower sub-network can be directly utilized by all the rest deeper sub-networks. In this manner, the key features are well preserved without significant increment of computing complexity. The information flows (including the key features) can be effectively reused across all sub-networks so that the deblurring effect can be improved while maintaining a short running time and a small model size.

### C. Loss Function

Different from other *coarse-to-fine* approaches, we design different depth networks to process the *coarse-to-fine* features. Instead of evaluating the Mean Square Error (MSE) loss at each sub-network, since all shallower sub-networks directly connect the deepest sub-network, the put-forth approach evaluates the MSE loss only at the deepest sub-network. The design of MSE loss in this approach is to measure the averaged squared errors of all pixels on the deblurred image and the ground truth. The smaller value of MSE means the closer deblurred-image to the ground truth image. Since the images are true color images that contain the red, green, and blue three channels, we need to compute the MSE of the deblurred image and the ground truth on these three channels. We then take the average of the results of the MSEs on three channels to obtain the final loss of an entire image. We denote the output of deblurred image and the ground truth (i.e., the sharp image) by $D$ and $S$, respectively. The red, green, and blue channels of $D$ and $S$ are denoted by $r$, $g$ and $b$, respectively. Then, the MSE loss values of each channel denoted by $\mathcal{L}_r$, $\mathcal{L}_g$ and $\mathcal{L}_b$

are defined as follows,

$$\mathcal{L}_r = \frac{1}{H \times W} \sum_{i=1}^{H \times W} (D_r^i - S_r^i)^2, \tag{6}$$

$$\mathcal{L}_g = \frac{1}{H \times W} \sum_{i=1}^{H \times W} (D_g^i - S_g^i)^2, \tag{7}$$

$$\mathcal{L}_b = \frac{1}{H \times W} \sum_{i=1}^{H \times W} (D_b^i - S_b^i)^2, \tag{8}$$

where $\mathcal{L}_r$, $\mathcal{L}_g$ and $\mathcal{L}_b$ are the square values of the $L_2$ norm of each channel error. The MSE Loss denoted by $\mathcal{L}_{rgb}$ form is the average of the MSE losses of the three channels as follows,

$$\mathcal{L}_{rbg} = \frac{1}{3}(\mathcal{L}_r + \mathcal{L}_g + \mathcal{L}_b). \tag{9}$$

The loss function of VDN denoted by $\mathcal{L}_{\text{VDN}}$ is an average of the loss values of all the samples, expressed as follows,

$$\mathcal{L}_{\text{VDN}} = \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}_{rgb}^i, \tag{10}$$

where $N$ is the number of samples in one training. Thus, we need to evaluate the loss function only at the deepest sub-network. The VDN follows the principle of residual learning, and the intermediate output captures image statistics at different depth sub-networks. Thanks to the *Stack Connection* among sub-networks, the original information in the first sub-network can be utilized in all other sub-networks repeatedly. Therefore, multi-level MSE loss is not applicable in the VDN model. Moreover, computing MSE loss only at the last sub-network can reduce the computing cost.

### D. Variant Depth Effect

The deblurring effect becomes better with the increased depth of sub-networks. Fig. 4 illustrates the variant depth effect of the VDN, where $D_i$ indicates the output result of each sub-network of VDN. The output image of each sub-network becomes sharper, less sparse, and richer in color with the increased depth after the level-by-level processing, especially when observing the magnified views.

The network contains finer information when its depth is deeper. As explained in the early parts, the VDN model uses several ResBlocks in the four-level structure. The depth of each sub-network becomes deeper with the increased number of ResBlocks. With the increased number of ResBlocks, the depth of each sub-network becomes deeper so as to achieve the coarse-to-fine deblurring effect via the variant depth sub-networks. In particular, we develop a four-level VDN-2345, where $n = 2$ in the first sub-network, $n = 3$ in the second sub-network, $n = 4$ in the third sub-network, $n = 5$ in the fourth sub-network. Fig. 4 shows the intermediate results (i.e., the outputs) of the four sub-networks of VDN-2345. The deblurring effects of deeper sub-networks become sharper in comparison to the shallower sub-networks. Thus, the final sub-network outputs the deblurred image.

## IV. Experimental Results

This section evaluates the effectiveness of the proposed VDN for the traffic (vehicle)-image deblurring. We provide a detailed performance comparison between the VDN models and SOTA methods in both quantitative and qualitative evaluations. The experiments are performed on a workstation with an i7-7700k CPU and an NVIDIA RTX 2080TI GPU. For a fair comparison, experiments are performed in the benchmark datasets with the same training configurations, and all tests are conducted on the same machine (unless noted otherwise).

### A. Datasets

We mainly evaluate the put-forth model on three representative datasets in the experiments:

1) **GoPro dataset** [16] contains 3,214 sharp-blurry-image (SBI) pairs extracted from 33 sequences at the resolution of $720 \times 1280$. For a fair comparison, we follow similar settings to [16], which used the training dataset (containing 2,103 SBI pairs) and the testing dataset (containing 1,111 SBI pairs) for the experiment.

2) **HIDE dataset** [46] contains 8,422 SBI pairs, extensively annotated with 65,784 bounding boxes. The images selected from 31 high-fps (frames per second) videos consist of realistic outdoor scenes with various numbers, poses, and human appearances at various distances. The images are divided into two categories: 1) the objects with the long-shot depth (i.e., HIDE I) and 2) objects with the close-ups depth (i.e., HIDE II). In HIDE II, the foreground images of people have undergone more significant motions than HIDE I. Thus, it will be more challenging to process blurred images in HIDE II than HIDE I. The experiments use 2,025 SBI pairs for the testing, where 1,063 SBI pairs are from HIDE I dataset and 962 SBI pairs are from HIDE II dataset.

3) **Need for Speed (NFS) dataset** [47] consists of 100 pairs of videos captured with a high frame rate. We use NFS mainly for qualitative evaluation.

We implement the VDN model on the PyTorch platform. We preprocess the training datasets with several data-augmentation techniques to mitigate the effects of overfitting. In particular, we first rotate the images in the range of [90°, 360°) randomly. Secondly, we gamma-correct the images and adjust the saturation of the image color with a random saturation factor ranging in $(0.5, 1.5]$. Thirdly, we crop the processed images by $256 \times 256$ pixels randomly. Last but not the least, we use Adaptive Moment Estimation Optimizer for the optimization, where the size of the mini-batch is 8. The initial learning rate is 0.0001 degrading a half per 500 epochs. All the training parameters are initialized with the Xavier method [48]. Thereafter, the above parameters are fixed for all experiments.

### B. Quantitative Evaluation

To evaluate the deblurring effect of the put-forth VDN, the experiments compare the VDN on the GoPro dataset with SOTA deblurring methods, including scale-recurrent

deblurring (SRNDeblur) [17], DeblurGAN-v2 [43], dynamic scene deblurring (DSDeblur) [49], hierarchical multi-patch deblurring (Stack(4)-DMPHN) [45], and multi-temporal recurrent neural networks for progressive non-uniform single image deblurring (MTRNN) [50]. We consider the following main comparison metrics:

1) Peak Signal-to-Noise Ratio (PSNR),
2) Structural Similarity Index Measure (SSIM),
3) model size,
4) running time.

*1) Definition of PSNR & SSIM:* In particular, both higher PSNR scores and higher SSIM scores mean the model performing better in the image deblurring task. The equations of PSNR and SSIM are given as Eq. (12) and Eq. (13), respectively.

Given a reference image $R$ and a test image $T$, both of size $m \times n$, we get MSE of $R$ and $T$ as:

$$\mathrm{MSE}(R,T) = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[R(i,j) - T(i,j)]^2, \tag{11}$$

Then, the equation of PSNR follows,

$$\mathrm{PSNR}(R,T) = 10\log_{10}\big(\mathrm{MAX}_R{}^2/\mathrm{MSE}(R,T)\big), \tag{12}$$

where $\mathrm{MAX}_R$ is the biggest pixel value of the reference image $R$. For the 8-bit binary images, $\mathrm{MAX}_R = 255$. Thus, a higher score of PSNR indicates a higher image quality [51]. For the image deblurring task, a higher score of PSNR means a better deblurring effect.

The SSIM is calculated according to the comparison of three factors between the reference image $R$ and the test image $T$: 1) luminance denoted by $l(R,T)$, 2) contrast denoted by $c(R,T)$, 3) structure denoted by $s(R,T)$. Let $\mu_r, \mu_t$ denote the mean values of $R$ and $T$, respectively, and $\sigma_r, \sigma_t$ denote the variance values of $R$ and $T$, respectively, and $\sigma_{rt}$ denotes the covariance of $R$ and $T$. Let $c_1, c_2$ be two constants, where $c_3 = c_2/2$. Then, we have $c_1 = (k_1 P)^2$, $c_2 = (k_2 P)^2$, where $k_1 = 0.01, k_2 = 0.03$, and $P = 255$ for the 8-bit binary images. The term SSIM is defined as follows:

$$\begin{aligned}
\mathrm{SSIM}(R,T) &= l(R,T)c(R,T)s(R,T),\\
l(R,T) &= \frac{2\mu_r\mu_t + c_1}{{\mu_r}^2 + {\mu_t}^2 + c_1},\\
c(R,T) &= \frac{2\sigma_r\sigma_t + c_2}{{\sigma_r}^2 + {\sigma_t}^2 + c_2},\\
s(R,T) &= \frac{\sigma_{rt} + c_3}{\sigma_r\sigma_t + c_3}.
\end{aligned} \tag{13}$$

The SSIM is used to measure the similarity between two images [51] so that the higher score of SSIM means a better deblurring effect in the image deblurring task.

For a fair comparison, we run all compared methods on the same platform. Experimental results are obtained after running SOTA models (executing their source codes or pre-trained models). Note that the running time is the average time of deblurring 1,111 SBI pairs from the test set of the GoPro dataset.

TABLE I
QUANTITATIVE RESULTS OF VDNS AND SOTA MODELS ON GOPRO DATASET

| Models | PSNR | SSIM | Size | Running-time |
|---|---|---|---|---|
| SRNDeblur [17] | 30.20 | 0.9334 | 33.6MB | 814ms |
| DeblurGAN-v2 [43] | 29.08 | 0.9183 | 244.5MB | **142ms** |
| DSDeblur [49] | 30.96 | 0.9421 | 49.8MB | 805ms |
| Stack(4)-DMPHN [45] | 31.39 | 0.9477 | 86.9MB | 637ms |
| MTRNN [50] | 31.13 | 0.9447 | **10.6MB** | 485ms |
| VDN-1234 | 31.17 | 0.9453 | 28.8MB | 265ms |
| VDN-2345 | **31.53** | **0.9487** | 36.5MB | 340ms |

*2) SOTA Comparison on GoPro and HIDE:* TABLE I and TABLE II show the quantitative results on the GoPro dataset and HIDE dataset, respectively. TABLE I lists the scores of PSNR, SSIM, model size, and running time among SRNDeblur, DeblurGAN-v2, DSDeblur, Stack(4)-DMPHN, MTRNN, and two representatives of the VDN (VDN-2345 and VDN-1234), on GoPro dataset. TABLE II lists the scores of PSNR and SSIM among these five methods and the proposed VDN-2345, on the HIDE dataset. It is obvious that the proposed VDN-1234 and VDN-2345 outperform the compared deblurring methods on both the GoPro dataset and the HIDE dataset.

In TABLE I, the highest scores are highlighted in bold. It is obvious that the put-forth model of VDN-2345 obtains the best scores 31.53 in the term of PSNR and 0.9487 in the term of SSIM, respectively. The second best one is Stack(4)-DMPHN, however, its model size is more than twice the VDN-2345. In addition, the VDN-1234 has the second smallest size, only 28.8 MB, which is nearly 10% of that of DeblurGAN-v2. Meanwhile, the VDN-1234 achieves even higher PSNR and SSIM values than DeblurGAN-v2. Although MTRNN [50] has the smallest model size, its running time is much higher than ours. Moreover, the VDN-1234 also achieves a good performance in terms of running time, i.e., the running time of VDN-1234 is 265 ms, much smaller than those of other models except DeblurGAN-v2 [43]. Although DeblurGAN-v2 achieves 142 ms (i.e., the best) on running time, its model size is nearly ten times of the VDN-1234. It is worth mentioning that DeblurGAN-v2 [43] costs a shorter time but a larger model size while MTRNN [50] costs a longer time with a smaller size on the contrary. In contrast, the VDN-1234 has obvious advantages over the SOTA models since its running time is only 265 ms and its model size is only 28.8 MB. The methods in [17] and [49] use multi-scale inputs to increase the receptive field to restore blurry images and spend more computing time than DeblurGAN-v2 [43].

Although the VDN-2345 ranks the third place (VDN-1234 ranks the second place) on the model size and the running time, its model size and running time are still very small. It implies that VDN models can achieve superior performance, i.e., the best deblurring effects with small model size and a short running time. The superior performance of the VDN model mainly owes to the *coarse-to-fine* architecture

TABLE II

QUANTITATIVE RESULTS OF PSNR AND SSIM ON HIDE DATASET

| Models | HIDE I | | HIDE II | |
|---|---|---|---|---|
| | PSNR | SSIM | PSNR | SSIM |
| SRNDeblur [17] | 29.18 | 0.9119 | 27.46 | 0.8907 |
| DeblurGAN-v2 [43] | 28.29 | 0.8960 | 26.65 | 0.8722 |
| DSDeblur [49] | 29.98 | 0.9234 | 28.14 | 0.9021 |
| Stack(4)-DMPHN [45] | 29.80 | 0.9247 | 28.34 | 0.9099 |
| MTRNN [50] | 29.98 | 0.9265 | 28.24 | 0.9081 |
| VDN-2345 | **30.03** | **0.9269** | **28.48** | **0.9117** |

TABLE III

COMPARISON WITH SOTA METHODS ON TRAFFIC IMAGES IN TERMS OF PSNR AND SSIM

| Models | GoPro (309) | | HIDE (1177) | |
|---|---|---|---|---|
| | PSNR | SSIM | PSNR | SSIM |
| SRNDeblur [17] | 30.18 | 0.9103 | 25.07 | 0.8539 |
| DeblurGAN-v2 [43] | 28.68 | 0.8879 | 24.57 | 0.8419 |
| DSDeblur [49] | 30.96 | 0.9215 | 25.80 | 0.8686 |
| Stack(4)-DMPHN [45] | 31.45 | 0.9286 | 26.32 | 0.8818 |
| MTRNN [50] | 31.05 | 0.9251 | 26.12 | 0.8785 |
| VDN-2345 | **31.65** | **0.9311** | **26.34** | **0.8824** |

to achieve the outstanding deblurring effect and the *stack connection* to reduce the running time and model size via reusing information flows across different levels of networks.

TABLE II shows results on HIDE dataset. We only choose the VDN-2345 for the comparison because it has excellent performance on GoPro dataset. The highest scores are highlighted in bold. The best results are all from the VDN-2345. All the models perform worse on HIDE dataset than those on GoPro. In particular, PSNR values of the SOTA models are less than 30.0 though the VDN-2345 achieves the best among all the models. Moreover, Stack(4)-DMPHN performs the second best on GoPro and performs not the same well on HIDE datasets in terms of PSNR and SSIM. For example, the PSNR value is 29.80 less than 29.98 obtained by DSDeblur and MTRNN. The smallest model size achieved by MTRNN [50] performs not well as its performance of SSIM on the HIDE II dataset, i.e., less than 0.9099 achieved by Stack(4)-DMPHN [45]. Meanwhile, the best running time achieved by DeblurGAN-v2 [43] performs worst among all methods as its PSNR and SSIM values are both the lowest. However, the VDN-2345 always performs the best on both HIDE I and HIDE II datasets. The quantitative results on the GoPro dataset and HIDE dataset demonstrate that the VDN can achieve superior performances, i.e., restoring the high-quality deblurring images with a small model size and a short running time.

*3) SOTA Comparison on Traffic Images:* In order to further evaluate the deblurring effect of the proposed VDN being applied in ITS, we delicately select some traffic scenario images from GoPro and HIDE datasets for further comparison.


Fig. 5. Results of deblurred images in city traffic scenarios. The top and bottom rows show the blurred images and ground truth images, respectively. From left to right the first three images are from the GoPro dataset and the fourth image is from the HIDE dataset. The rest rows from top to bottom show the deblurred results by SRNDeblur [17], DeblurGAN-v2 [43], DSDeblur [49], Stack(4)-DMPHN [45], MTRNN [50], and the VDN-2345, respectively.


Fig. 6. Results of deblurred images in highway traffic scenarios. The first column shows the blurry images in NFS dataset, and the second and third columns show the deblurred images processed by the put-forth VDN and the ground truth, respectively.

In particular, we pick up 309 SBI pairs of traffic images from the GoPro dataset and 1,177 SBI pairs about traffic images from the HIDE dataset. TABLE III shows the quantitative

results compared with the SOTA deblurring methods. Most of the models perform better on traffic images than those on the whole GoPro dataset when comparing the values of PSNR of the GoPro column in TABLE I and TABLE III. Moreover, the VDN-2345 obtains the best performance in both datasets. In particular, the PSNR value of the VDN-2345 reaches 31.65 which is the best among all models. On the other hand, the PSNR and SSIM of the VDN-2345 are 26.34 and 0.8824, respectively. The proposed VDN model achieves the best scores when processing blurry traffic images. The results imply that the VDN is well suitable for ITS applications, especially considering the resource limitation of ITS facilities (e.g., cameras, sensors, and IoT nodes).

### C. Qualitative Evaluation

*1) SOTA Comparison in City Traffic Scenarios:* We further conduct a qualitative evaluation of the VDN model with a comparison of other SOTA models. Fig. 5 shows the visual comparison results. In particular, Fig. 5 shows the deblurring effect for processing the traffic images in city traffic scenarios. The bottom and top rows indicate sharp and blurry images, respectively. From top to bottom, the rest rows show the deblurred images processed by SRNDeblur [17], DeblurGAN-v2 [43], DSDeblur [49], Stack(4)-DMPHN [45], MTRNN [50], and the proposed model of VDN-2345. The first three-columns images, from left to right, are images of the GoPro dataset and the fourth-column images are from HIDE dataset. It is obvious that license plates at the top row are too blurry to be recognized. After processing by the VDN-2345 and the SOTA methods, the texts on license plates become clearer than the original blurry images. However, only the deblurred images done by the VDN-2345 are the closest to the ground-truth images among all the methods. Especially, "T" is difficult to recognize in the images deblurred by SRNDeblur [17], DeblurGAN-v2 [43], and MTRNN [50], at the first column. Moreover, in the third column, the deblurred result of the VDN clearly shows the two driving persons (the closest to the ground truth image), though it is difficult to recognize that there are two persons in the images deblurred by SRNDeblur [17], DeblurGAN-v2 [43], DSDeblur [49], Stack(4)-DMPHN [45], MTRNN [50]. In the fourth column, there are still ghost effects in the images deblurred by DSDeblur [49] and Stack(4)-DMPHN [45]. By contrast, the result of the VDN is closer to the ground truth than those deblurred by SRNDeblur [17], DeblurGAN-v2 [43], and MTRNN [50].

This promising result implies that the VDN model is quite feasible for ITS scenarios such as smart parking systems. The third column shows a scenario of motorcycle drivers and passengers moving at a high speed when the face image of the person is too blurred to be recognized. The deblurred image by the VDN-2345 is also quite close to the ground-truth image. It is helpful to recognize the facial image of the person when applying the deblurring model in ITS applications, such as autonomous vehicles and transportation safety. The fourth column shows the scenario of pedestrians, when a pedestrian runs the red light, it is necessary to gain information from the facial image of the pedestrian. The blurred image taken by the camera is difficult to recognize. However, the deblurred image processed by the VDN-2345 model is quite close to the sharp image (i.e., the ground-truth image), thereby being used for further analysis, such as face recognition.

TABLE IV
QUANTITATIVE RESULTS OF VDNS AND SOTA METHODS ON GOPRO DATASET TESTED ON MEC PLATFORM

| Models | Running-time (Power model) | |
|---|---|---|
| | 15W (6-CORE) | 10W (2-CORE) |
| SRNDeblur [17] | 12.090s | 14.896s |
| DSDeblur [49] | 11.542s | 12.166s |
| Stack(4)-DMPHN [45] | 16.027s | 20.654s |
| VDN-1234 | **3.920s** | **4.399s** |
| VDN-2345 | 4.954s | 5.524s |

*2) SOTA Comparison in Highway Scenarios:* Fig. 6 shows the traffic conditions in highway scenarios. The first column shows the blurry images chosen from NFS dataset [47]. The second and third columns show the deblurred images processed by the VDN and the ground truths, respectively. Traffic signs are blurred in the first column. The blurry images may bring challenges in ITS applications, such as autonomous vehicles. For example, it may cause danger if a blurred traffic sign cannot be recognized by an autonomous vehicle that is moving at a high speed on a highway. The traffic signs in the deblurred images by the VDN-1234 can be clearly recognized (quite close to the ground-truth images), as shown in the second column of Fig. 6. Moreover, it is also critical for an image-processing time as well as the model size in the autonomous-vehicle scenario while the VDN model can well fulfill the critical requirement due to the lowest running time and the compacted model size.

## V. DISCUSSION

Mobile Edge Computing (MEC) has been increasingly applied in ITS [52], [53], [54]. We deploy the VDN to an MEC platform, NVIDIA Jetson Xavier NX Developer Kit (NJXDK), to evaluate its deblurring effect. NJXDK is one of the smallest Artificial Intelligence supercomputers of the MEC systems [55]. In particular, featuring an integrated GPU of 384-CORE NVIDIA Volta, CPU of 6-CORE NVIDIA Carmel ARM, and the memory of 8 GB 128-bit LPDDR4x, this MEC platform provides a high-performance accelerated software stack of NVIDIA CUDA-X™ and supports PyTorch library.

We test the VDN-1234 and VDN-2345 and SOTA methods including SRNDeblur [17], DSDeblur [49], Stack(4)-DMPHN [45] on the NJXDK MEC platform. Since MTRNN [50] needs a large memory and DeblurGAN-v2 [43] needs other plug-ins, they cannot be directly executed on the MEC platform. Thus, MTRNN and DeblurGAN-v2 are not considered in this experiment. We use the trained models for the comparison since the trained models can run on the MEC platform. We test images from GoPro dataset [16] and NFS dataset [47]. Since the values of PSNR and SSIM

are not variant with different platforms, experiments only need to compare the running time. The results are shown in TABLE IV. In particular, we evaluate two power models of MEC, i.e., 15W (6-CORE) power model and 10W (2-CORE) power model. It is obvious that two VDN models outperform other compared methods in the running time on both two power models. The VDN-1234 only spends 3.920 seconds and 4.399 seconds in terms of average running time on 15W (6-CORE) power model and 10W (2-CORE) power model, respectively. Moreover, the running time is increased from 15W (6-CORE) model to 10W (2-CORE) power model for the compared three methods, especially Stack(4)-DMPHN [45], while the running time of two VDN models is only slightly increased.

## VI. Conclusion

To address the challenges of the image-deblurring model applied in ITS applications, we designed a VDN model in a *coarse-to-fine* manner. The VDN model has a small model size and short computing time, thereby being beneficial for ITS applications. The proposed variant-depth sub-networks use residual modules with different depths in the four-level network. From the first (the shallowest) sub-network to the fourth (the deepest) one, the depth of the sub-networks becomes deeper, and the deblurring information processed by each sub-network becomes richer. In this *coarse-to-fine* way, the VDN processed fine-grained feature maps, consequently obtaining sharper restored images. Therefore, the VDN outperformed SOTA methods on deblurring quality. It is worth mentioning that we also devised a new connection (namely *Stack Connection*) connecting all sub-networks to fully use the deblurring information from each sub-network to reduce parameters and computing time. Moreover, we conducted several experiments to evaluate the VDN framework. Experimental results demonstrated that the proposed VDNs outperformed SOTA image-deblurring methods on several representative datasets. In particular, the proposed model performed the best PSNR and SSIM scores while maintaining the shortest running time and the smallest model size.

## References


[1] E. Wang et al., "Deep learning-enabled sparse industrial crowdsensing and prediction," *IEEE Trans. Ind. Informat.*, vol. 17, no. 9, pp. 6170–6181, Sep. 2021.

[2] X. Yu and S. Shen, "An integrated decomposition and approximate dynamic programming approach for on-demand ride pooling," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 9, pp. 3811–3820, Sep. 2020.

[3] X. Shi, H. Qi, Y. Shen, G. Wu, and B. Yin, "A spatial–temporal attention approach for traffic prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 8, pp. 4909–4918, Aug. 2021.

[4] Z. Zheng, Y. Yang, J. Liu, H.-N. Dai, and Y. Zhang, "Deep and embedded learning approach for traffic flow prediction in urban informatics," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 10, pp. 3927–3939, Oct. 2019.

[5] X. Xu et al., "Service offloading with deep Q-network for digital twinning empowered Internet of Vehicles in edge computing," *IEEE Trans. Ind. Informat.*, vol. 18, no. 2, pp. 1414–1423, Feb. 2022.

[6] K. Liu, W. Wang, R. Tharmarasa, and J. Wang, "Dynamic vehicle detection with sparse point clouds based on PE-CPD," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 5, pp. 1964–1977, May 2018.

[7] W. Wang et al., "Vehicle trajectory clustering based on dynamic representation learning of Internet of Vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 6, pp. 3567–3576, Jun. 2021.

[8] G. Chen et al., "Pseudo-image and sparse points: Vehicle detection with 2D LiDAR revisited by deep learning-based methods," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 12, pp. 7699–7711, Dec. 2021.

[9] A. Mehra, M. Mandal, P. Narang, and V. Chamola, "ReViewNet: A fast and resource optimized network for enabling safe autonomous driving in hazy weather conditions," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 7, pp. 4256–4266, Jul. 2021.

[10] Z. Liu et al., "Robust target recognition and tracking of self-driving cars with radar and camera information fusion under severe weather conditions," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 6640–6653, Jul. 2022.

[11] J. Jia, "Single image motion deblurring using transparency," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–8.

[12] Q. Shan, J. Jia, and A. Agarwala, "High-quality motion deblurring from a single image," *ACM Trans. Graph.*, vol. 27, no. 3, p. 73, 2008.

[13] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce, "Non-uniform deblurring for shaken images," *Int. J. Comput. Vis.*, vol. 98, no. 2, pp. 168–186, 2012.

[14] C. J. Schuler, M. Hirsch, S. Harmeling, and B. Schölkopf, "Learning to deblur," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 7, pp. 1439–1451, Sep. 2016.

[15] T. M. Nimisha, A. K. Singh, and A. N. Rajagopalan, "Blur-invariant deep learning for blind-deblurring," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4752–4760.

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

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

[18] P. Wieschollek, M. Hirsch, B. Scholkopf, and H. P. A. Lensch, "Learning blind motion deblurring," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 231–240.

[19] J. Pan, W. Ren, Z. Hu, and M.-H. Yang, "Learning to deblur images with exemplars," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 6, pp. 1412–1425, Jun. 2019.

[20] Y. Tang, W. Gong, X. Chen, and W. Li, "Deep inception-residual Laplacian pyramid networks for accurate single-image super-resolution," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 5, pp. 1514–1528, May 2020.

[21] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4700–4708.

[22] M. Naeem, W. Ejaz, M. Iqbal, F. Iqbal, A. Anpalagan, and J. J. P. C. Rodrigues, "Efficient scheduling of video camera sensor networks for IoT systems in smart cities," *Trans. Emerg. Telecommun. Technol.*, vol. 31, no. 5, pp. 1–13, May 2020.

[23] R. Q. Mínguez, I. P. Alonso, D. Fernández-Llorca, and M. Á. Sotelo, "Pedestrian path, pose, and intention prediction through Gaussian process dynamical models and pedestrian activity recognition," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 5, pp. 1803–1814, May 2018.

[24] A. Rasouli and J. K. Tsotsos, "Autonomous vehicles that interact with pedestrians: A survey of theory and practice," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 3, pp. 900–918, Mar. 2020.

[25] A. Belhadi, Y. Djenouri, G. Srivastava, D. Djenouri, J. C.-W. Lin, and G. Fortino, "Deep learning for pedestrian collective behavior analysis in smart cities: A model of group trajectory outlier detection," *Inf. Fusion*, vol. 65, pp. 13–20, Jan. 2021.

[26] Y. Wang, S. Zhao, R. Zhang, X. Cheng, and L. Yang, "Multi-vehicle collaborative learning for trajectory prediction with spatio-temporal tensor fusion," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 1, pp. 236–248, Jan. 2021.

[27] K. S. Boujemaa, I. Berrada, K. Fardousse, O. Naggar, and F. Bourzeix, "Toward road safety recommender systems: Formal concepts and technical basics," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 6, pp. 5211–5230, Jun. 2022.

[28] W. Balid, H. Tafish, and H. H. Refai, "Intelligent vehicle counting and classification sensor for real-time traffic surveillance," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 6, pp. 1784–1794, Jun. 2018.

[29] A. M. Ranwa, F. Bilal, and Q. Alejandro, "Cooperative evaluation of the cause of urban traffic congestion via connected vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 1, pp. 59–67, Jan. 2019.

[30] W. Ma, W. Liu, X. Luo, K. McAreavey, Y. Jiang, and J. Ma, "A Dempster–Shafer theory and uninorm-based framework of reasoning and multiattribute decision-making for surveillance system," *Int. J. Intell. Syst.*, vol. 34, no. 11, pp. 3077–3104, Nov. 2019.

[31] F. Bock, S. Di Martino, and A. Origlia, "Smart parking: Using a crowd of taxis to sense on-street parking space availability," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 2, pp. 496–508, Feb. 2020.

[32] R. Martin Nieto, A. Garcia-Martin, A. G. Hauptmann, and J. M. Martinez, "Automatic vacant parking places management system using multicamera vehicle detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 3, pp. 1069–1080, Mar. 2019.

[33] S. Zang, M. Ding, D. Smith, P. Tyler, T. Rakotoarivelo, and M. A. Kaafar, "The impact of adverse weather conditions on autonomous vehicles: How rain, snow, fog, and hail affect the performance of a self-driving car," *IEEE Veh. Technol. Mag.*, vol. 14, no. 2, pp. 103–111, Jun. 2019.

[34] A. Gupta, N. Joshi, C. Lawrence Zitnick, M. Cohen, and B. Curless, "Single image deblurring using motion density functions," in *Proc. Eur. Conf. Comput. Vis.*, K. Daniilidis, P. Maragos, and N. Paragios, Eds., 2010, pp. 171–184.

[35] Y. Lu, G. Lu, J. Li, Y. Xu, Z. Zhang, and D. Zhang, "Multiscale conditional regularization for convolutional neural networks," *IEEE Trans. Cybern.*, vol. 52, no. 1, pp. 444–458, Jan. 2022.

[36] H. Wu, X. Chen, P. Li, and Z. Wen, "Automatic symmetry detection from brain MRI based on a 2-channel convolutional neural network," *IEEE Trans. Cybern.*, vol. 51, no. 9, pp. 4464–4475, Sep. 2021.

[37] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[38] C. Ren, X. He, Y. Pu, and T. Q. Nguyen, "Learning image profile enhancement and denoising statistics priors for single-image super-resolution," *IEEE Trans. Cybern.*, vol. 51, no. 7, pp. 3535–3548, Jul. 2021.

[39] S. Nah, S. Son, and K. M. Lee, "Recurrent neural networks with intra-frame iterations for video deblurring," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 8102–8111.

[40] C. Wang et al., "Self-supervised multiscale adversarial regression network for stereo disparity estimation," *IEEE Trans. Cybern.*, vol. 51, no. 10, pp. 4770–4783, Oct. 2021.

[41] Z. Zhong, J. Li, D. A. Clausi, and A. Wong, "Generative adversarial networks and conditional random fields for hyperspectral image classification," *IEEE Trans. Cybern.*, vol. 50, no. 7, pp. 3318–3329, Jul. 2020.

[42] O. Kupyn, V. Budzan, M. Mykhailych, D. Mishkin, and J. Matas, "DeblurGAN: Blind motion deblurring using conditional adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8183–8192.

[43] O. Kupyn, T. Martyniuk, J. Wu, and Z. Wang, "DeblurGAN-V2: Deblurring (orders-of-magnitude) faster and better," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 8877–8886.

[44] H. Zhu et al., "Single-image dehazing via compositional adversarial network," *IEEE Trans. Cybern.*, vol. 51, no. 2, pp. 829–838, Feb. 2012.

[45] H. Zhang, Y. Dai, H. Li, and P. Koniusz, "Deep stacked hierarchical multi-patch network for image deblurring," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5978–5986.

[46] Z. Shen et al., "Human-aware motion deblurring," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 5572–5581.

[47] H. K. Galoogahi, A. Fagg, C. Huang, D. Ramanan, and S. Lucey, "Need for speed: A benchmark for higher frame rate object tracking," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1134–1143.

[48] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. 13th Int. Conf. Artif. Intell. Statist.*, in Proceedings of Machine Learning Research, vol. 9, Y. W. Teh and M. Titterington, Eds. Sardinia, Italy, May 2010, pp. 249–256. [Online]. Available: https://proceedings.mlr.press/v9/glorot10a.html

[49] H. Gao, X. Tao, X. Shen, and J. Jia, "Dynamic scene deblurring with parameter selective sharing and nested skip connections," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3848–3856.

[50] D. Park, D. U. Kang, J. Kim, and S. Y. Chun, "Multi-temporal recurrent neural networks for progressive non-uniform single image deblurring with incremental temporal training," in *Computer Vision—ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham, Switzerland: Springer, 2020, pp. 327–343.

[51] A. Hore and D. Ziou, "Image quality metrics: PSNR vs. SSIM," in *Proc. 20th Int. Conf. Pattern Recognit.*, Aug. 2010, pp. 2366–2369.

[52] Y. Cao, X. Zhang, B. Zhou, X. Duan, D. Tian, and X. Dai, "MEC intelligence driven electro-mobility management for battery switch service," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 7, pp. 4016–4029, Jul. 2021.

[53] P. Dai, K. Hu, X. Wu, H. Xing, F. Teng, and Z. Yu, "A probabilistic approach for cooperative computation offloading in MEC-assisted vehicular networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 2, pp. 899–911, Feb. 2022.

[54] I. Sorkhoh, C. Assi, D. Ebrahimi, and S. Sharafeddine, "Optimizing information freshness for MEC-enabled cooperative autonomous driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 13127–13140, Aug. 2022.

[55] P. Grzesik and D. Mrozek, "Metagenomic analysis at the edge with Jetson Xavier NX," in *Proc. Int. Conf. Comput. Sci.* Cham, Switzerland: Springer, 2021, pp. 500–511.

**Qian Wang** (Student Member, IEEE) received the B.Eng. degree in electronic information engineering from Yangtze University, Jingzhou, China, and the M.Eng. degree in educational technology from the Zhejiang University of Technology, Zhejiang, China. She is currently pursuing the Ph.D. degree in computer technology and application with the Faculty of Information Technology, Macau University of Science and Technology.

Her research interests include deep learning, image processing, and AI painting.

**Cai Guo** (Student Member, IEEE) received the Ph.D. degree from the School of Computer Science and Engineering, Macau University of Science and Technology, Macao SAR.

He is currently working with Hanshan Normal University, Chaozhou, China. His research interests include deep learning, image processing, and computer vision.

**Hong-Ning Dai** (Senior Member, IEEE) received the Ph.D. degree in computer science and engineering from the Department of Computer Science and Engineering, The Chinese University of Hong Kong.

He is currently an Associate Professor with the Department of Computer Science, Hong Kong Baptist University. He has published more than 200 papers in top-tier journals/conferences and received more than 12000 citations. His research interests include the Internet of Things, big data analytics, and blockchain. He is also a Senior Member of the Association for Computing Machinery. He has served as an Associate Editor/Editor for IEEE COMMUNICATIONS SURVEYS AND TUTORIALS, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, IEEE TRANSACTIONS ON INDUSTRIAL CYBER-PHYSICAL SYSTEMS, and *Ad Hoc Networks* (Elsevier).

**Min Xia** (Senior Member, IEEE) received the B.S. degree in industrial engineering from Southeast University, China, in 2009, the M.S. degree in precision machinery and precision instrumentation from the University of Science and Technology of China, China, in 2012, and the Ph.D. degree in mechanical engineering from The University of British Columbia, Canada, in 2017.

He is currently an Assistant Professor with the School of Engineering, Lancaster University, U.K. He has led 11 research projects in the U.K., Canada, and Japan, with total funding of U.S. $10 million. His research interests include smart manufacturing, machine diagnostics and prognostics, deep neural networks, and process monitoring and optimization. He has served various editorial roles, including an Associate Editor for IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENTS.