Boosting Night-Time Scene Parsing With Learnable Frequency

Zhifeng Xie\textsuperscript{a,}, Sen Wang\textsuperscript{a,}, Ke Xu\textsuperscript{a,}, Zhizhong Zhang\textsuperscript{a,}, Xin Tan\textsuperscript{a,}, Member, IEEE, Yuan Xie\textsuperscript{a,}, and Lizhuang Ma\textsuperscript{a,}, Member, IEEE

Abstract—Night-Time Scene Parsing (NTSP) is essential to many vision applications, especially for autonomous driving. Most of the existing methods are proposed for day-time scene parsing. They rely on modeling pixel intensity-based spatial contextual cues under even illumination. Hence, these methods do not perform well in night-time scenes as such spatial contextual cues are buried in the over-/under-exposed regions in night-time scenes. In this paper, we first conduct an image frequency-based statistical experiment to interpret the day-time and night-time scene discrepancies. We find that image frequency distributions differ significantly between day-time and night-time scenes, and understanding such frequency distributions is critical to NTSP problem. Based on this, we propose to exploit the image frequency distributions for night-time scene parsing. First, we propose a Learnable Frequency Encoder (LFE) to model the relationship between different frequency coefficients to measure all frequency components dynamically. Second, we propose a Spatial Frequency Fusion module (SFF) that fuses both spatial and frequency information to guide the extraction of spatial context features. Extensive experiments show that our method performs favorably against the state-of-the-art methods on the NightCity, NightCity+ and BDD100K-night datasets. In addition, we demonstrate that our method can be applied to existing day-time scene parsing methods and boost their performance on night-time scenes. The code is available at https://github.com/wangsenn/FDLNet.

Index Terms—Night-time vision, scene parsing, frequency analysis.

I. INTRODUCTION

Scene parsing is a fundamental task in computer vision with many downstream applications, such as autonomous driving [1], human parsing [2], and image inpainting [3]. Most representative scene parsing methods [4], [5], [6], [7], [8] are proposed for day-time scenes. However, while night-time may contribute to half of the total working hours (e.g., in autonomous driving), these existing methods do not work well in night-time scenes due to the day-time/night-time scene discrepancies (see Figure 1(a)). Meanwhile, although there are some methods [9], [10], [11], [12], [13] proposed to transfer the day-time domain knowledge to the night-time domain for scene parsing through domain adaptation, they still cannot achieve practical performances due to the less resolved domain discrepancies.

Recently, Tan et al. [14] propose the first large-scale night-time scene dataset (NightCity). They also propose an exposure-guided network for night-time scene parsing (NTSP). Deng et al. [15] propose the NightLab, which further boosts the performance of NTSP by learning the image lighting variation and mining hard segmented regions.

However, all these methods typically rely on modeling pixel-intensity-based contextual features, which are not necessarily reliable under uneven night-time lighting conditions. On the other hand, we note that some style transfer-based segmentation methods [12], [13] assume that the low-level spectrum represents scene lighting information. Hence, two questions are raised: Can image frequency distributions represent the day-time/night-time domain discrepancies? And are all frequency components important for NTSP?

To answer the aforementioned two questions, we first conduct an image-frequency based analysis. We first analyze image-level frequency distributions by randomly select one day-time image from the Cityscapes [16] and one night-time image from the NightCity [14] (Figure 1(a)). We use the Discrete Cosine Transform (DCT) to compute the spectrum of images as in [17]. Following the JPEG compression process [18], the image is divided into multiple $8 \times 8$ blocks. Then,
we calculate the mean value of spectrograms of all blocks as shown in Figure 1(b). While the frequency distribution of day-time image does differ from that of night-time image and such difference mainly comes from the low frequency components, we can see that night-time images do have different high frequency distribution from day-time image.

We further analyze the local regions of the night-time image where under- and over-exposures happen (marked with orange and blue boxes Figure 1(a)). For the corresponding comparing regions of day-time image we select the objects with the same semantics (i.e., cars). We compute the spectrograms of those regions as shown in Figure 1(c). We can see that the high frequency distribution of day-time image tends to have less peaks due to its relatively even lighting condition, while that of night-time image tends to have more peaks. This demonstrates that high frequency distribution differences reveal the lighting discrepancies of different domains.

Furthermore, we perform quantitative experiments at the dataset level to demonstrate our observation. To better analyze the frequency difference, we divide the spectrogram into four parts, as shown in Figure 2(a). First, we calculate the mean values of the spectrogram in each frequency region, and then calculate the variance of the mean values of each frequency region of all images in the day/night dataset separately. We show the results in Figure 2(b) that the variance of the night-time scenes in each region is larger than that of the day-time scenes, which indicates that the difference of the dataset-level frequency information for night-time scenes is also significant. This motivates us to design a network for NTSP that is learnable for all frequency information to adjust the frequency components dynamically.

In this paper, we propose a novel Frequency Domain Learning Network (FDLNet), which first deals with the NTSP in the frequency domain. Specifically, we first propose a Learnable Frequency Encoder (LFE) which fully exploits all frequency components generated by DCT to adjust the channel response of different frequency components dynamically. Since the frequency distribution of different night-time images is diverse, the LFE can adaptively adjust the channel response of frequency components, so the weights of frequency components are unique for each image. Then, we propose a Spatial Frequency Fusion module (SFF), which fuses the spatial features and frequency features in channel-wise. We use both spatial and frequency information to guide the extraction of spatial context features for NTSP. We conduct extensive experiments on night-time datasets (including NightCity, NightCity+ and BDD100K-night), showing that our method plays favorably against state-of-the-art methods. Besides, our method can be easily applied to existing state-of-the-art day-time segmentation methods [5], [6], [7], [19] to adapt them for NTSP. In sum, our main contributions are:

1) We interpret the scene lighting discrepancies between day-time and night-time scenes with the image frequency distributions. We show that a full understanding of image frequency distributions is crucial to NTSP. We propose a novel Frequency Domain Learning Network (FDLNet) for NTSP.

2) We propose the Learnable Frequency Encoder (LFE), to dynamically adjust the channel response of all frequency components. We propose the Spatial Frequency Fusion module (SFF), which fuses the spatial features and frequency features in channel-wise. We use both spatial and frequency information to model spatial contexts by a fusion of spatial and frequency features.

3) We show that our method can easily be applied to state-of-the-art day-time scene parsing methods to boost their performances for NTSP.

II. RELATED WORK

A. Scene Parsing

Scene parsing aims to assign each pixel with its class label. Long et al. [4] propose the first fully convolutional
network (FCN) to extract deep features for scene parsing. Later, many methods such as PSPNet [5] and DeepLab [6], [20], [21] are proposed to aggregate more spatial features by expanding their reception fields. In order to obtain more effective spatial features, a variety of attention mechanisms have been studied in scene parsing. In [22], Point-wise Spatial Attention is proposed to associate the information of each location with that of other locations. Self-attention mechanism is introduced in DANet [23] and OCNet [24] to capture contextual information. C4Net [7] and Axial-DeepLab [25] apply the Non-local module [26] to model long-range spatial contextual information. Recently, transformer-based methods are proposed to model global contexts for scene parsing. SETR [8] uses the transformer layers to form the encoder for extracting the global context information. Swin Transformer [19] uses the sliding windows with information exchange mechanism to reduce the computational complexity of transformer, while capturing global information. Strudel et al. [27] propose a fully transformer architecture with a Mask Transformer as the decoder to generate class masks. Xie et al. [28] propose to fuse multi-level features without positional encoding in the encoding stage.

Meanwhile, there are also some methods proposed to encode prior knowledge into scene parsing. HANet [29] models the height distribution statistics of object categories, based on which they propose to learn height-driven attention. SANet [30] factorizes the scene parsing task into two sub-tasks of pixel classification and pixel grouping, and leverages pixel grouping to aggregate contextual information to enhance pixel classification. ISNet [31] learns both image level and semantic level contextual features to model inter- and intra-class correlation for scene parsing. STLNet [32] uses the proposed Quantization and Counting Operator to leverage the low-level texture features for scene parsing. In [33], contextual information beyond a single image is modeled via their proposed MCIBI by dynamically building dataset-level semantic features during training.

B. Night-Time Scene Parsing

Due to the lack of large-scale night-time datasets for network learning, previous NTSP methods have to resort to semi-supervised learning [34] or domain adaption. For example, ADVENT [35] proposes an entropy loss in the segmentation scene to reduce the loss from the source domain to the target domain. DAFFormer [36] builds a transformer-based segmentation framework for domain adaptation learning. HRDA [37] proposes a multi-resolution unsupervised domain adaptation training method that can adapt to small objects and fine detail segmentation. Some methods require the day-time and night-time image pairs as input [9], [10], [11], [12], and reduce the domain gap via the bridge of twilight time [9], a generic uncertainty-aware annotation and evaluation method in curriculum framework [10], a re-weighting strategy during adversarial training [11], the inter-domain style adaptation and intra-domain gradual self-training [12]. Most recently, Tan et al. [14] propose a large-scale real night-time dataset and an exposure-guided network to learn robust semantic features. Deng et al. [15] propose the NightLab, which further boosted the performance of NTSP by learning the image lighting variation and mining hard-segmented regions. Wang et al. [38] divide the network into two sub-networks of illumination enhancement and image segmentation to segment night-time images.

All previous methods rely on pixel-intensity based spatial contextual information, which may not be reliable due to the existence of over- and under-exposures in night-time scenes. In this paper, we study the NTSP problem from the image frequency perspective, showing that an understanding of frequency distributions facilitates contextual information modeling significantly.

C. Deep Learning in the Frequency Domain

A line of methods explores deep learning in the image frequency domain for network acceleration and compression. Chen et al. [39] convert the weights of convolutional filters to the frequency domain, and use a hash function to randomly group frequency parameters to hash buckets and compress models by sharing parameters in hash buckets. Considering the spatial redundancy in most filters in CNNs, Liu et al. [40] propose a frequency-domain dynamic pruning scheme that exploits spatial correlation to compress the model. Ehrlich et al. [41] propose a general approach to perform residual network inference and learning in the JPEG transform domain, where they reconstruct the residual network and take compressed images as input to skip the costly decompression step for faster image processing. Xu et al. [17] propose a learning-based frequency selection method and the network accepts frequency information as input and reduce the size of input images.

Frequency-based methods are also exploited for dense-prediction tasks. Liu et al. [42] propose a new multi-level wavelet CNN model, which uses wavelet transform to balance the receptive field size and computational efficiency for image restoration. Zou et al. [43] design a content-aware anti-aliasing module to adapt to downsampling and reduce the loss of high-frequency details caused by downsampling. The method is effective in image classification, semantic segmentation, instance segmentation and other fields. Qin et al. [44] propose a frequency-channel attention network to improve existing network structures by using frequency-domain priors and achieve state-of-the-art performance on image classification, object detection, and instance segmentation tasks. Qiu et al. [45] propose a joint space-time-frequency domain transformer structure for compressed video super-resolution. Xu et al. [13] perform day-to-night image style transfer in the Fourier domain and utilized domain adaptation for night-time scene parsing. Li et al. [46] decouples the body (low frequency) and edge (high frequency) features of the image to optimize the boundary details for scene parsing.

However, the existing frequency domain methods are immature for the NTSP task and night-time images have different frequency features compared with other kinds of images. Some image segmentation methods use fixed frequency components [17], [44] or do not explicitly model frequency features [46] or simply utilize low-level spectrograms for image preprocessing [13] but all frequency components of night-time
images are important as we have observed in the Introduction. Different from previous work, we propose to model and learn the whole image frequency distributions for NTSP.

III. THE PROPOSED METHOD

A. Overview

In this paper, we present a novel method that models frequency distribution to facilitate the night-time scene parsing. Figure 3 shows the pipeline of the proposed method. Given an RGB image $I$, the backbone encodes the images into a spatial feature map, denoted as $f_{\text{spatial}}$. Then, we compute frequency features $f_{\text{freq}}$ by transforming $f_{\text{spatial}}$ into the frequency domain with Discrete Cosine Transform. To fully exploit frequency information, we first propose a Learnable Frequency Encoder (LFE). This module re-weights frequency feature $f_{\text{freq}}$ based on the contribution of each frequency component. Second, we propose a novel spatial frequency fusion module to fuse the spatial $f_{\text{spatial}}$ and frequency $f_{\text{freq}}$ information in channel-wise. After fusion, a standard segmentation head is attached to produce the final parsing results.

B. 2D Discrete Cosine Transform

We employ Discrete Cosine Transform (DCT) to transfer the spatial feature map to frequency domain. First, we simply review the principle of DCT. The basic function of the two-dimensional discrete cosine transform $B$ is:

$$B_{u,v} = \cos \left( \frac{(2u+1)\pi}{2N} \right) \cos \left( \frac{(2v+1)\pi}{2N} \right),$$

where $u$ and $v$ are the horizontal and vertical frequency components, respectively. $N$ is the size of an image block, and $(x, y)$ represents the spatial locations of the image block.

Then the two-dimensional discrete cosine transform can be formulated as:

$$F(u, v) = c(u)c(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) B_{x,y}^{u,v},$$

where $F(u, v)$ is the 2D DCT frequency spectrum, $u, v \in [0, 1, \cdots, n-1]$, and $f(x, y)$ is a two-dimensional vector element of $N \times N$ in the spatial domain, $x, y \in [0, 1, \cdots, N-1]$. $c(u)$ and $c(v)$ are compensation factors, written as:

$$c(u), c(v) = \begin{cases} \sqrt{\frac{1}{N}}, & u, v = 0 \\ \sqrt{\frac{2}{N}}, & u, v \neq 0. \end{cases}$$

Following [44], we utilize DCT to record the frequency information in the channel dimension. Given input spatial feature map $f_{\text{spatial}} \in \mathbb{R}^{C \times H \times W}$, where $C, H$ and $W$ denote the channel dimension, height and width, respectively. According to the rules of image compression and coding, we reconstruct the size of $f_{\text{spatial}}$ into $N \times N$. Then, $f_{\text{spatial}}$ is divided into multiple parts in the channel dimension to obtain $f_{\text{spatial}}^{i} \in \mathbb{R}^{C \times H \times W}$, where $n$ is the total number of frequency components. Thus, we can obtain each frequency component $f_{\text{freq}}^{i}$ by its corresponding spatial feature component $f_{\text{spatial}}^{i}$ using 2D DCT function $DCT^{i}$:

$$f_{\text{freq}}^{i} = DCT^{i} \left( f_{\text{spatial}}^{i}(x, y) \right)$$

$$= c(u)c(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f_{\text{spatial}}^{i}(x, y) B_{x,y}^{u,v}$$

s.t. $i \in [0, 1, \cdots, n-1]$.

![Figure 3](image-url)
After that, the multi-spectral frequencies vector \( V_{freq} \in \mathbb{R}^C \) is defined as:
\[
V_{freq} = \text{cat} \left( [f_{freq}^0, f_{freq}^1, \cdots, f_{freq}^{n-1}] \right),
\]
where \( \text{cat} \) denotes concatenate operation.

C. Learnable Frequency Encoder

Unlike day-time scenes, the frequency distribution of night images is more discrete (see Figure 1 and 2). Simply utilizing a fixed number of frequency components is not an ideal solution for night-time scene parsing. Hence, we propose the Learnable Frequency Encoder (LFE) to learn the importance of each frequency component. To dynamically adjust each frequency component, a Learnable Frequency Component Convolutional layer (L FCC) is used to convert the entire multi-spectral frequency vector \( V_{freq} \) into the weight of each frequency component \( W \), as:
\[
W = \text{softmax} \left( \text{L FCC} \left( V_{freq} \right) \right),
\]
where \( \text{L FCC} \) includes a \( 1 \times 1 \) convolutional layer and a batchnorm layer. For training stability, we constrain the weights of \( \text{L FCC} \) to be positive and sum them to 1 by a \( \text{softmax} \) function. The \( V_{freq} \) is reshaped to the size of \( n \times C \times 1 \times 1 \) and \( W \in \mathbb{R}^{n \times C \times 1 \times 1} \), where each weight of the \( 1^2 \) channel corresponds to one frequency component \( f_{freq}^i \in \mathbb{R}^{C \times 1 \times 1} \). This operation can be expressed as:
\[
f_{freq}^i = w^i \cdot f_{freq}, \tag{7}
\]
where \( w^i \) is one channel of \( W \) corresponding to each frequency component \( f_{freq}^i \). Then we calculate the re-weight multi-spectral frequencies vector \( V'_{freq} \) as follows:
\[
V'_{freq} = WV_{freq} = \text{cat} \left( w_0^0 f_{freq}^0, w_1^1 f_{freq}^1, \cdots, w_{n-1}^{n-1} f_{freq}^{n-1} \right) = \text{cat} \left( f_{freq}^0, f_{freq}^1, \cdots, f_{freq}^{n-1} \right). \tag{8}
\]
We use \( n \) to group the consecutive channels of frequency vector \( V_{freq} \) and the output of filter \( W \) adjusts the weight of each frequency component \( f_{freq}^i \) based on \( V_{freq} \). By multiplying \( w \) and \( V_{freq} \) element-wise, the encoder is learnable to predict the weight of each frequency component. Finally, the output of the encoder is the re-weight frequency feature \( V'_{freq} \) which is rectified at the channel dimension.

Discussion: We note that there are some methods [17], [44] proposed to model image frequency information but only select top \( k \) frequency components to represent the whole image. However, as shown in Figure 1 and 2, high-frequency components still contain important information due to the uneven lighting conditions of night-time scenes. Our module models the whole image frequency distribution and adjusts their weights dynamically. The experiment in Table V shows that dynamically modeling the whole image frequency distribution facilitates the NTSP performance.

D. Spatial Frequency Fusion

Modeling the frequency distribution helps the network understand the scene illumination. We then use the learned frequency features to guide the network to model spatial context features for night-time scene parsing. Specifically, we propose the Spatial Frequency Fusion module (SFF) to fuse features from two different domains. First, we employ a spatial context aggregation module to enhance the extraction of spatial features \( f_{spatial} \) and then utilize a convolutional layer to transform the \( f_{spatial} \) into spatial representations \( R_s \in \mathbb{R}^{C \times \frac{H}{8} \times \frac{W}{8}} \). Meanwhile, the re-weight frequency feature \( V'_{freq} \) is fed into a convolutional layer to reduce the dimensionality to generate frequency representations \( R_f \). After that \( R_f \in \mathbb{R}^{C \times 1 \times 1} \) is extended to \( R_f \in \mathbb{R}^{C \times \frac{H}{8} \times \frac{W}{8}} \) so as to keep the same shape with \( R_s \). Then both \( R_f \) and \( R_s \) are reshaped to \( \mathbb{R}^{C \times D} \), where \( D = \frac{H}{8} \times \frac{W}{8} \). We conduct matrix multiplication between the transpose of reshaped \( R_f \) and \( R_s \), and apply a \( \text{Softmax} \) layer to calculate the affinity map. The affinity operation is then defined as:
\[
A(i, j) = \frac{\exp \left( R_i^s \otimes (R_j^s)^T \right)}{\sum_{i=1}^{C} \exp \left( R_i^s \otimes (R_j^s)^T \right)}, \tag{9}
\]
where \( A(i, j) \) indicates the effect of \( i^{th} \) channel in the spatial representations \( R_s \) on the \( j^{th} \) channel in the frequency representations \( R_f \) and \( \otimes \) denotes matrix multiplication. \( A \) is the affinity map calculated over the channel dimension. After that, the final fused representation \( R_{final} \) is calculated as follows:
\[
R_{final} = \alpha \left( \text{permute} \left( A \otimes R_s \right) \right) + R_s, \tag{10}
\]
where \( \text{permute} \) reshapes the result of \( A \otimes R_s \) to \( C \times \frac{H}{8} \times \frac{W}{8} \) and \( \alpha \) is a scale parameter to reduce gradient instability. Note that each channel of \( R_{final} \) is the weighted sum of all channels through spatial and frequency features, and effectively captures the long-term dependencies between spatial and frequency domains.

E. Loss Function

We use the standard cross-entropy loss to measure the errors between the network predictions and ground truth labels. In addition, since the high-frequency boundary information is an important cue for scene parsing, we also incorporate edge loss during training. Unlike previous methods [46], [47] that learn edge information in the spatial domain, which are not reliable in night-time scenes due to their complex lighting conditions, we propose to learn edge information in the frequency domain.

Let \( s_{i,c} \) and \( s'_{i,c} \) be the ground-truth and prediction results of the \( i^{th} \) pixel of class \( c \), respectively. \( L_{edge} \) focus on the semantic edge regions of \( s_{i,c} \) as:
\[
L_{edge} = - \sum_i \sum_c I_{b_i} \cdot (s_{i,c} \log s'_{i,c}), \tag{11}
\]
where \( L_{edge} \) represents cross-entropy loss on semantic edge regions. \( b_i \) is the ground-truth semantic edge of the \( i^{th} \) pixel.
and the $I_{by}$ represents indicator function that semantic edge region in the ground-truth $s_{i,c}$.

The overall loss $L$ can be defined as:

$$L = \lambda_1 L_{\text{seg}} + \lambda_2 L_{\text{edge}},$$

(12)

where $L_{\text{seg}}$ is a standard cross-entropy loss. $\lambda_1$ and $\lambda_2$ are two hyperparameters that control the weighting between the losses.

IV. EXPERIMENTS

A. Datasets and Implementation Details

Datasets: NightCity [14] is the first large-scale labeled night-time scene dataset for training and validation. NightCity+ [15] refines some labeling errors in the validation set of NightCity [14]. There is another night-time dataset, BDD100K-night, which selects night-time images with their labels from the BDD100K [48] as described in [14] and [15]. Finally, we also test our model on a day-time dataset Cityscapes [16] to verify its generalization ability.

1) NightCity: It includes 4,297 finely annotated images, of which 2,998 images are used for training, and 1,299 images are used for validation. The dataset labels are compatible with Cityscapes and contain 19 categories, and the resolution of the images is 512×1024.

2) NightCity+: NightCity+ updates the validation set of NightCity by correcting the labeling errors, and resizes the resolution of the image to 1024×2048.

3) BDD100K-Night: It has 320 images in the training set and 34 images in the validation set. It also has 19 categories same as Cityscapes and the image resolution is 720×1280.

4) Cityscapes: It contains 5,000 annotated images, including 2,975 images for training, 500 images for validation, and 1,525 images for testing. The label contains 19 classes, and the resolution of the images is 1024×2048.

Implementation Details: The PyTorch framework is employed to implement our network. In the training phase, our model uses stochastic gradient descent (SGD) optimizer and a poly learning strategy with $\left(1 - \frac{\text{iter}}{\text{total\_iter}}\right)^{0.9}$. We set the initial learning rate and weight decay coefficients to 5e-3 and 5e-4, respectively. Moreover, we set the batch size to 8, and the crop size is 384×768. We conduct experiments on one TITAN RTX GPU. For data augmentation, we use random scaling with ratio sampled in the range of (0.5, 2.0), random horizontal flip, crop, and Gaussian blur as in [5]. And the training time is set to 260 epochs. For evaluation, we use multi-scale inference [49] with ratios of [0.5, 0.75, 1.0, 1.25, 1.5, 1.75]. We utilize the dilated residual network [50] as the backbone with an output stride of 1/8. In the process of SFF, to reduce the amount of computation, we use the projection function to reduce the number of channels to 512. We set $\lambda_1 = 1$ and $\lambda_2 = 0.01$ according to Table I.

B. Comparison on the NightCity and NightCity+

To verify the effectiveness of our method, we train our model and other state-of-the-art methods on the NightCity train set and evaluate them on the NightCity validation set and the NightCity+ validation set, respectively. For experimental comparison consistency, we rescale the NightCity+ validation set images to 512×1024.

1) Methods for Comparisons: We compare our model with state-of-the-art methods including EGNet [14] and NightLab [15] for Night-Time Scene Parsing (NTSP). PSPNet [5], DeepLabV3+ [6], DANet [23], CCNet [7], GSCNN [47], HANet [29], STER [8], UperNet [19] and SegFormer [28] for Day-Time Semantic Segmentation (DTSS). ADVENT [35], DAFomer [36] and HRDA [37] for Domain Adaptation (DA). However, some domain adaptation methods [9], [10], [11], [12] need the day-time locations corresponding to the night-time ones as network input to reduce the domain adaptation loss, and the acquisition conditions of this kind of data are harsh, they are not included for comparison. We report the performances of EGNet and HANet from [14] and NightLab from [15]. PSPNet, DeepLabV3+, DANet and CCNet are trained with the same configurations as ours. Other methods use their official code and configurations for training. Since our method can be applied to day-time segmentation methods for NTSP, we report the results of our model based on PSPNet [5], DeepLabV3+ [6], CCNet [7] and UperNet [19].

Quantitative Comparison: From Table II, we can see that the day-time and domain adaptation methods cannot achieve satisfying results due to the large gap between day and night scenes, but our proposed method can successfully adapt the day-time model to the night-time scenes. Furthermore, our model based on the PSPNet [5] outperforms the EGNet with a margin of 1.41%. To gain better results, we utilize our model on a stronger baseline DeepLabV3+ [6] and obtain 1.39% improvement on NightCity and 1.95% improvement on NightCity+. We also use a multi-scale strategy during inference and achieve a performance of 56.79%, which outperforms the NightLab based on DeepLabV3+ with a margin of 0.58%. Moreover, our model based on the transformer-based method UperNet [19] achieves 55.54% and outperforms the baseline model with a margin of 0.61% on NightCity and 0.83% on NightCity+. Noting that the resolution is different between ours and NightLab. Our method requires smaller resolution inputs which reduces computation but achieves higher performance. We also use the same resolution as NightLab and our model performs better than Nightlab. The results show that our model improves the day-time models to adapt to NTSP and shows superior performance and good generalization.

Qualitative Comparison: Figure 4 quantitatively compares the prediction results of our model with state-of-the-art methods for DTSS and NTSP. Since NightLab does not show segmentation results on NightCity, we compare our results with EGNet and UperNet. The lighting conditions of night-time

| Method | epoch | batchsize | cropsize | $\lambda_1$ | $\lambda_2$ | mIoU(%) |
|--------|-------|-----------|----------|-------------|-------------|---------|
| PDLNet (PSPNet) | 400   | 16        | 384x384  | 1           | 0.5         | 46.92   |
| -      | 400   | 16        | 384x384  | 1           | 0.01        | 47.83   |
| -      | 340   | 10        | 512x512  | 1           | 0.01        | 47.01   |
| -      | 300   | 8         | 384x768  | 1           | 0.1         | 51.40   |
| -      | 260   | 8         | 384x768  | 1           | 0.01        | 53.21   |

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
TABLE II
COMPARISON WITH STATE-OF-THE-ARTS ON NIGHTCITY AND NIGHTCITY+. DTSS REPRESENTS DAY-TIME SEMANTIC SEGMENTATION, NTSP REPRESENTS NIGHT-TIME SCENE PARSING AND DA REPRESENTS DOMAIN ADAPTATION. † STANDS FOR USING ONLINE HARD EXAMPLE MINING DURING TRAINING [51]. * STANDS FOR MULTI-SCALE INFERENCE THE BEST RESULTS ARE HIGHLIGHTED IN BOLD

| Method | Years | Original Task | Backbone | Resolution | mIoU(%) |
|--------|-------|---------------|----------|------------|---------|
|        |       |               |          |            | NightCity | NightCity+ |
| PSPNet [5] | CVPR 2017 | DTSS | ResNet-101 | 512 × 1024 | 51.02 | 52.24 |
| DeepLabV3+ [6] | ECCV 2018 | DTSS | ResNet-101 | 512 × 1024 | 51.99 | 53.26 |
| DANet [23] | CVPR 2019 | DTSS | ResNet-101 | 512 × 1024 | 50.81 | 52.47 |
| CCNet [7] | ICCV 2019 | DTSS | ResNet-101 | 512 × 1024 | 49.81 | 50.94 |
| GSCNN [47] | ICCV 2019 | DTSS | WideResNet38 | 512 × 1024 | 48.92 | - |
| HANet [29] | CVPR 2020 | DTSS | ResNet-101 | 512 × 1024 | 51.1 | - |
| SETR [8] | CVPR 2021 | DTSS | ViT-L | 512 × 1024 | 43.11 | - |
| UperNet [19] | ICCV 2021 | DTSS | Swin-T | 512 × 1024 | 54.93 | 56.24 |
| SegFormer [28] | NeurIPS 2021 | DTSS | MIT-B5 | 512 × 1024 | 46.28 | - |
| EGNNet [14] | TIP 2021 | NTSP | ResNet-101 | 512 × 1024 | 51.8 | - |
| NightLab (DeepLabV3+) [15]* | CVPR 2022 | NTSP | ResNet-101 | 1024 × 2048 | - | 56.21 |
| ADVENT [35] | CVPR 2019 | DA | ResNet-101 | 512 × 1024 | 24.43 | 25.67 |
| DAFormer [36] | CVPR 2022 | DA | MIT-B5 | 512 × 1024 | 33.10 | 35.65 |
| HRDA [37] | ECCV 2022 | DA | MIT-B5 | 512 × 1024 | 38.38 | 41.46 |
| FDLNet (PSPNet)† | - | NTSP | ResNet-101 | 512 × 1024 | 53.21 | 54.25 |
| FDLNet (CCNet)†† | - | NTSP | ResNet-101 | 512 × 1024 | 51.00 | 52.27 |
| FDLNet (DeepLabV3+) | - | NTSP | ResNet-101 | 512 × 1024 | 53.39 | 55.19 |
| FDLNet (DeepLabV3+)†* | - | NTSP | ResNet-101 | 512 × 1024 | 55.42 | 56.79 |
| FDLNet (UperNet) | - | NTSP | Swin-T | 512 × 1024 | 55.54 | 57.04 |
| FDLNet (UperNet)* | - | NTSP | Swin-T | 1024 × 2048 | - | 57.39 |

Fig. 4. Qualitative comparison on NightCity. Our advantages are highlighted by orange boxes.

images make the frequency distribution quite different. Adjusting the lighting condition cannot allow the network to learn the frequency information, which makes the segmentation effect unsatisfactory. However, our model can handle this problem well. Particularly, in the first row, our model can identify areas of detail, such as distant buildings and poles. In the second row, our model gives more complete poles than EGNNet and more complete trees than UperNet. In the third row, our model segments a more complete building. In the last row, EGNNet cannot segment objects such as traffic lights, traffic signs and poles due to overexposure. UperNet cannot segment these objects completely. But our model can recognize small objects with complete fineness. These results demonstrate the superior performance of our proposed model on NTSP.

C. Comparison on the BDD100K-Night

The previous NTSP methods, EGNNet [14] and NightLab [15], selected the night image in BDD100K and named it BDD100K-night. We followed them and tested our approach. To verify the effectiveness of our method, we compare our model with state-of-the-art methods PSPNet [5], DeepLabV3+ [6], DANet [23], CCNet [7], STER [8], UperNet [19],
SegFormer [28], and AGLN [52] for day-time semantic segmentation. EGNet [14] for Night-time scene parsing. Since the codes of NightLab [15] is not available, it is not included for comparison. The results are reported in Table III. We can see that our model based on DeepLabV3+ achieves the best performance of 26.46% among CNN-based methods and our model based on UperNet achieves the best performance of 30.87%, which shows the superiority and generality of our proposed method. Moreover, we conduct experiments on BDD100K as the baseline. A simply way to leverage frequency features is to integrate them into the network for 120 epochs, as shown in Figure 5. We find that selecting 8 frequency components brings about 0.37% improvement (4th row), which shows the importance of edge supervision.

D. Model Analysis

1) Ablation Study: To verify the effectiveness of different network components, we conduct five ablation studies.

a) Ablation studies on the number of frequency components: The number of frequency components is one of the important factors affecting model performance. The network extracts image features and compresses the information into channel representations, so we use DCT to compress spatial features into \( N \times N \) blocks to extract frequency features. Due to the limitation of channel numbers, \( N \) could be 2, 4, 8, 16, or 32. We use the ResNet-50 as the backbone and train the network for 120 epochs, as shown in Figure 5. We find that selecting \( 8 \times 8 \) frequency components obtain the best performance. In other experiments, we set the number of frequency components to \( 8 \times 8 \).

b) Ablation studies on baseline model: We take the PSPNet [5] with OHEM [51] during the training process as the baseline. A simply way to leverage frequency features is to design a SENet-like [53] module as in FcaNet [44]. So we use the same way on LFE module to adjust the channel weights for the obtained frequency features and improve the performance from 51.90% to 52.14% (2nd row). Then, we leverage SFF module in the network to replace the SENet-like module, in order to introduce the spatial context features extracted by PPM [5]. The performance of the model is improved from 52.14% to 52.84% (3rd row). Incorporating the \( L_{edge} \) brings about 0.37% improvement (4th row), which shows the importance of edge supervision.

c) Ablation studies on learnable frequency encoder: To demonstrate the effectiveness of our proposed LFE module, we take PSPNet as a baseline and compare our method with two methods, one is using top-k components in [44] named (TOP), and the other is statically using all frequency components named (SA). We show the results in Table V, our learnable frequency encoder strategy achieves the best performance to varying degrees. Specifically, on the one hand,
SENet leverages linear layers to adjust the channel response without the spatial features of the image, which can achieve good results in image classification but is not suitable for the NTSP task of spatial pixel-level classification. On the other hand, SFF leverages convolutional layers to adjust the channel response including the spatial structure information from two different domains (spatial and frequency) and outperforms the former by 1.15%. The scale parameter $\alpha$ reduces gradient instability during training since adjusting the frequency component channel responses with information from all channels is a computationally expensive task. $\alpha$ can be changed incrementally to mitigate the drastic gradient changes, without the parameters, the performance of the model drops by 1.82%.

e) Improvements to day-time methods: Our method can be applied to the existing day-time methods to adapt them for the NTSP task. For consistency comparison, we modify PSP, DeeplabV3+, and CCNet by using our method with the same experimental settings. The results in Table VII show that our method improves the NTSP performance of existing day-time methods while introducing minimum computational overhead.

f) Compare with simpler networks of increased size: To demonstrate the effectiveness of our method, we use a simpler network FCN [4] as a baseline, and experiment with networks of different sizes. As shown in Table VIII, we can see that our method improve the original model in different sizes, i.e. 0.86% for ResNet-50, 1.34% for ResNet-101. Moreover, the performance of the network increased by 0.55% from resnet-50 to resnet-101, but the parameter amount increased by 18.99M (49.5M → 68.49M). Our method can improve the performance of resnet101 model by 1.34%, while the number of parameters is only increased by 1.71M. In other words, our approach is more efficient.

2) Compare with Frequency Domain Adaption: In some domain adaptation methods [12], [13], the frequency information is used to perform style transformation on the image to reduce the gap between the source and target domains, which is also suitable for the style transformation of day-time and night-time images. For comparison, we use NightCity as the source domain and Cityscapes as the target domain, so night-time images are transformed into daytime-like images by Frequency Domain Adaption (FDA) [13]. Then, we use DeeplabV3+ to obtain the prediction results. To gain more accurate results, we use the resized labels of the validation set of NightCity + (512×1024) for comparison.

a) Qualitative comparison: We report the results in Figure 6 and observe that transforming night-time images to day-time images using FDA can reduce the domain gap between them to a certain extent (blue boxes). However, simply replacing the frequency information of the two images often fails. For example, in the white boxes of the 5 and 8 rows, the transformed images are severely distorted, resulting in incomplete prediction results and chaotic boundaries, while our model uses learnable frequency information to guide the network to predict more complete predictions and the boundaries are clearer. More comparison results are highlighted by orange boxes. These visual results show that our proposed method is more efficient than directly preprocessing the image.

b) Quantitative comparison: For better comparison, we also report the results of DeeplabV3+ on the original NightCity, as shown in Table X. The performance of FDA (51.33%) is even worse than the original method (53.26%), which means that simply preprocessing the image with frequency information does not solve the NTSP problem well. Whereas our method introduces learnable frequency information into the model, the network learns the frequency distribution of night images and achieves better results.

3) Compare with Other Frequency Transform: DCT has a simpler structure than Fourier and wavelet transforms, and it is easier to learn frequency components. To demonstrate the superiority of DCT, we add the results using Fourier transform (FT) and wavelet transform (WT) as shown in Table IX. We can see that DCT outperforms FT and WT.

4) Compare With Day-Time Dataset Cityscapes: Our method focuses on night-time scene parsing, because
night-time scenes have two characteristics. First, the night scene contains information on all frequency components, including low-frequency areas with rich information and high-frequency areas with relatively little information. Moreover, the information contained in the high-frequency components of night-time images is richer than that of day-time images. Second, the frequency distribution of different night-time images is more different than day-time images, and the network can generate different component weights by learning each image.

To explore the difference between our method on night-time and day-time images, we train models on Cityscapes and NightCity and perform quantitative comparisons on three different validation sets of Cityscapes, NightCity and NightCity+. The training settings are the same, except the learning rates are 0.005 and 0.01 for NightCity and Cityscapes, respectively. From Table XI, we can see that our method achieves better results than baselines on both day-time and night-time datasets. The improvement is 1.31% on NightCity, 0.88% on NightCity+ and 0.23% on Cityscapes, which shows that our model is more effective on night-time images, and also shows the difference in frequency distribution between night-time and day-time images.

5) Model Perform under Noisy Labels: We verify the performance of the model under noisy labels with pseudo-labels generated by the unsupervised training method [34]. Specifically, our method improved the segmentation network in the unsupervised domain adaptation method (DAformer [36]).

| Method          | road | side | build | wall | fence | pole | light | sign | vege | terr | sky | pers | rider | car | truck | bus | train | moto | bicy | mIoU |
|-----------------|------|------|-------|------|-------|------|-------|------|------|------|-----|------|-------|-----|-------|-----|-------|------|------|------|
| DeeplabV3+      | 90.4 | 51.1 | 83.2  | 53.3 | 53.3  | 32.0 | 24.4  | 52.2 | 59.0 | 19.7 | 88.2| 52.2  | 25.2  | 82.8 | 64.9  | 73.8 | 59.1  | 10.2 | 34.7  | 53.26 |
| FDA (DeeplabV3+) | 90.4 | 50.6 | 82.4  | 53.5 | 52.2  | 30.1 | 23.3  | 49.0 | 56.9 | 20.6 | 87.4| 49.6  | 17.8  | 82.3 | 62.3  | 72.4 | 59.5  | 0    | 34.1  | 51.33 |
| PDLNet (PSFNet) | 90.5 | 50.8 | 83.2  | 53.9 | 53.1  | 28.6 | 23.8  | 51.6 | 59.1 | 21.1 | 87.9| 50.6  | 25.2  | 82.6 | 63.1  | 75.1 | 60.4  | 28.9 | 38.3  | 54.25 |
| PDLNet (DeeplabV3+) | 91.2 | 53.1 | 83.8  | 58.3 | 54.4  | 34.1 | 30.1  | 57.2 | 60.1 | 22.2 | 88.2| 55.9  | 27.6  | 84.6 | 61.3  | 73.8 | 58.8  | 29.2 | 44.0  | 56.20 |
TABLE XI
COMPARISON ON THE DAY-TIME DATASET

| Method     | NighCity | NighCity+ | Cityscapes |
|------------|----------|-----------|------------|
| PSPNet     | 51.90    | 53.37     | 72.19      |
| FDLNet(PSPNet) | 53.21    | 54.25     | 72.42      |
| Δ          | +1.31    | +0.88     | +0.23      |

TABLE XII
COMPARISON TO DOMAIN ADAPTATION

| Method     | NightCity | NightCity+ |
|------------|-----------|------------|
| DAFormer   | 33.10     | 35.65      |
| FDLNet (DAFormer) | 34.47     | 36.66      |

Fig. 7. Image-level LFE heatmap. (left) The source image. (right) In the LFE heatmap, the low frequency components are located in the upper left part and the high frequency components are located in the lower right part. Different images correspond to different frequency affinities.

results are shown in Table XII. Our method also improves the model performance of using pseudo labels, which shows that our model has good generalization.

E. Visual Analysis

To illustrate the capabilities of our proposed Learnable Frequency Encoder (LFE), we visualize the heatmap of LFE on the NightCity validation set.

1) Image-Level LFE: Our proposed LFE is able to dynamically adjust the weight of each frequency component, which means that there are differences of the frequency component affinity of each image. To illustrate this, we feed different images into the network to obtain the heatmap. Figure 7 shows that for different images, the weights of frequency components are also different. The frequency components with the largest weight in Figure 7(a) appear in the high-frequency regions, but the distribution of the frequency weights is relatively loose. While the frequency component with the largest weight is located in the low-frequency regions as shown in Figure 7(b), and the distribution of frequency weights is concentrated in low-frequency regions relatively. This shows the frequency distribution in night-time scenes is diverse as we observe in Figure 1 and Figure 2.

2) Dataset-Level LFE: To further analyze the LFE, we summed and averaged the LFE of all images in the validation set, resulting in a dataset-level LFE heatmap. As shown in Figure 8(a), we can see that our model prefers low-frequency components. The maximum weight of frequency component (*) is located in the low-frequency part, and the large weights are also generally concentrated in the low-frequency part, which proves that CNN prefers to select the low-frequency region with rich information in extracting features as [17] and [44]. However, the minimum weight of frequency component (#) is located in the low-frequency region rather than the high-frequency region, which reflects that the high-frequency information is equally important.

3) LFE on Semantic Edge Loss: Since we use the semantic edge loss $L_{edge}$, which focuses on the prediction of high-frequency related to semantic edges, so we visualize the LFE heatmap for analysis to demonstrate the effectiveness of semantic edge loss. Note that we use the same color numeric intervals in Figure 8(a) to visualize the results. Figure 8(a) shows the results w/o $L_{edge}$. The maximum weight (*) is located in the low-frequency part, and the minimum (#) is located in the high-frequency part. In contrast to method w/ $L_{edge}$, whose minimum weight (#) appears in the low-frequency region. This shows that semantic edge loss strengthens the attention of edge details to a certain extent. Furthermore, the model w/o $L_{edge}$ overall has a looser selection of frequency components compared to the model w/ $L_{edge}$, which indicates the semantic edge loss enforces the network to extract frequency features more efficiently and reduce information redundancy.

4) LFE on Day-Time Images: We also visualize the LFE applied to the day-time dataset Cityscapes. The comparison results are shown in Figure 8. We find that LFE has little effect on Cityscapes, and all frequency components have the same weight (Figure 8(c)). This also confirms our comparison of the frequency distribution of day-time and night-time datasets in the introduction that the frequency distribution of day-time images differs little, and the LFE cannot effectively learn and adjust its frequency components. Moreover, this partly explains why our method improves night-time datasets more than day-time datasets, as shown in Table XI.

V. CONCLUSION

In this paper, we propose a Frequency Domain Learning Network (FDLNet) to handle the frequency information distribution diversification of Night-Time Scene Parsing (NTSP). Specifically, the Learnable Frequency Encoder (LFE) adjusts the weights of frequency components generated by the DCT. Since high and low-frequency information is both important for NTSP, the encoder processes all frequency components information. Moreover, the encoder dynamically adjusts each
frequency component to adapt to changes in the frequency distribution of night images. Furthermore, the Spatial Frequency Fusion module (SFF) leverages information from two different domains to guide the network segmentation since only utilizing frequency information lacks spatial context features that are important for NTSP. Besides, our method allows a simple modification of the day-time model to adapt it to nighttime scenes. Our model achieves competitive performance on NightCity, NightCity+ and BDD100K-night against the state-of-the-art methods.

REFERENCES

[1] H. Fujiyoshi, T. Hirakawa, and T. Yamashita, “Deep learning-based image recognition for autonomous driving,” IATSS Res., vol. 43, no. 4, pp. 244–252, Dec. 2019.

[2] P. Li, Y. Xu, Y. Wei, and Y. Yang, “Self-correction for human parsing,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 44, no. 6, pp. 3260–3271, Jun. 2022.

[3] L. Liao, J. Xiao, Z. Wang, C.-W. Lin, and S. Sato, “Image inpainting guided by coherence priors of semantics and textures,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2021, pp. 6539–6548.

[4] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2015, pp. 3431–3440.

[5] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, “Pyramid scene parsing network,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 2881–2890.

[6] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, “Encoder-decoder with atrous separable convolution for semantic image segmentation,” in Proc. Eur. Conf. Comput. Vis. (ECCV), 2018, pp. 801–818.

[7] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, and W. Liu, “CCNet: Criss-cross attention for semantic segmentation,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 603–612.

[8] S. Zheng et al., “Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2021, pp. 6881–6890.

[9] D. Dai and L. V. Gool, “Dark model adaptation: Semantic image segmentation from daytime to nighttime,” in Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC), Nov. 2018, pp. 3819–3824.

[10] C. Sakaridis, D. Dai, and L. Van Gool, “Guided curriculum model adaptation and uncertainty-aware evaluation for semantic nighttime image segmentation,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 7374–7383.

[11] J. X. Wu, Z. Wu, H. Guo, L. Ju, and S. Wang, “DANNet: A one-stage domain adaptation network for unsupervised nighttime semantic segmentation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2021, pp. 15769–15778.

[12] Q. Xu, Y. Ma, J. Wu, C. Long, and X. Huang, “CDAda: A curriculum domain adaptation for nighttime semantic segmentation,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops (ICCVW), Oct. 2021, pp. 2962–2971.

[13] Y. Yang and S. Soatto, “FDA: Fourier domain adaptation for segmentation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 4085–4095.

[14] X. Tan, K. Xu, Y. Cao, Y. Zhang, L. Ma, and R. W. H. Lau, “Night-time scene parsing with a large real dataset,” IEEE Trans. Image Process., vol. 30, pp. 9085–9098, 2021.

[15] X. Deng, P. Wang, X. Lian, and S. Newsam, “NightLab: A dual-level architecture with hardness detection for segmentation at night,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2022, pp. 16938–16948.

[16] M. Cordts et al., “The cityscapes dataset for semantic urban scene understanding,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 3213–3223.

[17] K. Xu, M. Qin, F. Sun, Y. Wang, Y.-K. Chen, and F. Ren, “Learning in the frequency domain,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2020, pp. 1740–1749.

[18] G. K. Wallace, “The JPEG still picture compression standard,” Commun. ACM, vol. 34, no. 4, pp. 30–44, Apr. 1991.
Zhizhong Zhang received the Ph.D. degree in pattern recognition and intelligent systems from the Institute of Automation, Chinese Academy of Sciences (CAS), in 2020. He is currently an Associate Research Professor with the School of Computer Science and Technology, East China Normal University. His research interests include image processing, computer vision, machine learning, and pattern recognition.

Xin Tan (Member, IEEE) received the B.Eng. degree in automation from Chongqing University, China, in 2017, and the dual Ph.D. degrees in computer science from Shanghai Jiao Tong University and the City University of Hong Kong. He is currently an Associate Research Professor with the School of Computer Science and Technology, East China Normal University, China. His research interests include computer vision and deep learning. He serves as a Program Committee Member/Reviewer for CVPR, ICCV, ECCV, AAAI, IJCAI, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IEEE TRANSACTIONS ON IMAGE PROCESSING, and IJCV.

Zhifeng Xie received the Ph.D. degree in computer application technology from Shanghai Jiao Tong University, Shanghai, China. He was a Research Assistant with the City University of Hong Kong, Hong Kong. He is currently an Associate Professor with the Department of Film and Television Engineering, Shanghai University, Shanghai. He has published several works on CVPR, ECCV, IJCAI, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, and IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY. His current research interests include image/video processing and computer vision.

Yuan Xie received the Ph.D. degree in pattern recognition and intelligent systems from the Institute of Automation, Chinese Academy of Sciences (CAS), in 2013. He is currently a Full Professor with the School of Computer Science and Technology, East China Normal University. He has published more than 50 papers in major international journals and conferences, including IJCV, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, IEEE TRANSACTIONS ON CYBERNETICS, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, IEEE TRANSACTIONS ON MULTIMEDIA, NeurIPS, CVPR, and ECCV. His research interests include image processing, computer vision, machine learning, and pattern recognition. He received the Hong Kong Scholar Award from the Society of Hong Kong Scholars and the China National Postdoctoral Council in 2014. He has served as a reviewer for more than 15 journals and conferences.

Sen Wang received the B.Eng. degree in computer science and technology from Shanghai University, Shanghai, China, in 2021, where he is currently pursuing the M.Eng. degree with the Department of Film and Television Engineering. His current research interests include computer vision and deep learning.

Ke Xu received the dual Ph.D. degrees from the Dalian University of Technology and the City University of Hong Kong. He is currently with the City University of Hong Kong. His research interests include deep learning and image enhancement. He has served as a Program Committee Member/Reviewer for several CV and AI conferences and journals, including CVPR, ICCV, ECCV, NeurIPS, ICLR, ICML, AAAI, IJCV, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, and IEEE TRANSACTIONS ON IMAGE PROCESSING.

Lizhuang Ma (Member, IEEE) received the B.S. and Ph.D. degrees from Zhejiang University, China, in 1985 and 1991, respectively. He is currently a Distinguished Professor with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, China, and the School of Computer Science and Technology, East China Normal University, China. His research interests include computer vision, computer aided geometric design, computer graphics, scientific data visualization, computer animation, digital media technology, theory and applications for computer graphics, and CAD/CAM.