Learning Motion-Guided Multi-Scale Memory Features for Video Shadow Detection

Junhao Lin, Jiaxing Shen, Xin Yang, *Member, IEEE*, Huazhu Fu, *Senior Member, IEEE*, Qing Zhang, Ping Li, *Member, IEEE*, Bin Sheng, *Member, IEEE*, Liansheng Wang, *Member, IEEE*, and Lei Zhu, *Member, IEEE*

***Abstract*— Natural images often contain multiple shadow regions, and existing video shadow detection methods tend to fail in fully identifying all shadow regions, since they mainly learned temporal features at single-scale and single memory. In this work, we develop a novel convolutional neural network (CNN) to learn motion-guided multi-scale memory features to obtain multi-scale temporal information based on multiple network memories for boosting video shadow detection. To do so, our network first constructs three memories (i.e., a global memory, a local memory, and a motion memory) to combine spatial context and object motion for detecting shadows. Based on these three memories, we then devise a multi-scale motion-guided long-short transformer (MMLT) module to learn multi-scale temporal and motion memory features for predicting a shadow detection map of the input video frame. Our MMLT module includes a dense-scale long transformer (DLT), a dense-scale short transformer (DST), and a dense-scale motion transformer (DMT) to read three memories for learning multi-scale transformer features. Our DLT, DST, and DMT consist of a set of memory-read pooling attention (MPA) blocks and densely connect these output features of multiple MPA blocks to learn multi-scale transformer features since the scales of these output features are varied. By doing so, we can more accurately identify multiple shadow regions with different sizes from the input video. Moreover, we devise a self-supervised pretext task to pre-training the feature encoder for enhancing the downstream video shadow detection. Experimental results on three benchmark datasets show that our video shadow detection network quantitatively and qualitatively outperforms 26 state-of-the-art methods.**

***Index Terms*— Neural networks, video shadow detection.**

I. Introduction

SHADOWS are a ubiquitous feature in natural images, offering valuable cues for extracting scene geometry [1], [2], [3], [4], [5], estimating light directions, and determining camera locations and parameters [2]. Additionally, shadows have the potential to enhance a diverse range of image understanding tasks, including image segmentation [6], object detection [7], image editing [8], and object tracking [9]. The last decade has witnessed a growing interest in image shadow detection. Early methods addressed the shadow detection task in still single image by examining color and illumination priors [10], by developing data-driven approaches with hand-crafted features [11], [12], [13], or by learning deep discriminative features via diverse convolutional neural networks (CNNs) [14], [15], [16], [17], [18], [19], [20]. While image-based shadow detectors can be applied frame by frame to detect shadow pixels, their performance is often unsatisfactory due to the lack of consideration for temporal information from neighboring video frames.

Owing to an annotated video shadow detection dataset (i.e., ViSha [8]), much research attention [8], [21], [22], [23], [24], [25] on shadow detection has recently been shifted from single static images to dynamic videos. Hence, a common strategy to detect shadows from video data is to learn temporal features from adjacent video frames [8], [21], [22], [23], [24]. Chen et al. [8] annotated the first VSD dataset and presented a baseline network equipped with a dual gated co-attention module mechanism for enhancing correlations between video frames and a T-module for learning inter-video and intra-video features. Since then, researchers [21], [22], [23], [24] have noticed that temporal consistency is the central obstruction for accurately detecting shadows from video and usually

Manuscript received 21 November 2023; revised 29 March 2024 and 27 May 2024; accepted 27 May 2024. Date of publication 26 July 2024; date of current version 23 December 2024. This work was supported in part by the Hong Kong University of Science and Technology (HKUST)-Guangzhou (GZ) Joint Funding Program under Grant 2023A03J0671 and Grant 2024A03J0618; in part by the InnoHK funding launched by Innovation and Technology Commission, Hong Kong, SAR; in part by Guangzhou Industrial Information and Intelligent Key Laboratory Project under Grant 2024A03J0628; and in part by the Nansha Key Area Science and Technology Project under Grant 2023ZD003. This article was recommended by Associate Editor F. Díaz-de-María. *(Corresponding author: Lei Zhu.)*

Junhao Lin is with the Robotics and Autonomous Systems (ROAS) Thrust, The Hong Kong University of Science and Technology (Guangzhou), Nansha, Guangzhou, Guangdong 511400, China (e-mail: junhaolin@xmu.edu.cn).

Jiaxing Shen is with the Department of Computing and Decision Sciences, Lingnan University, Hong Kong (e-mail: jiaxingshen@LN.edu.hk).

Xin Yang is with the Department of Computer Science, Dalian University of Technology, Dalian 116024, China (e-mail: xinyang@dlut.edu.cn).

Huazhu Fu is with the Institute of High Performance Computing (IHPC), Agency for Science, Technology and Research (A*STAR), Singapore 138632 (e-mail: hzfu@ieee.org).

Qing Zhang is with the School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou 510006, China (e-mail: zhangqing.whu.cs@gmail.com).

Ping Li is with the Department of Computing, the School of Design, and the Research Institute for Sports Science and Technology, The Hong Kong Polytechnic University, Hong Kong (e-mail: p.li@polyu.edu.hk).

Bin Sheng is with the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: shengbin@sjtu.edu.cn).

Liansheng Wang is with the School of Informatics, Xiamen University, Xiamen 361005, China (e-mail: lswang@xmu.edu.cn).

Lei Zhu is with the Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Robotics and Autonomous Systems (ROAS) Thrust and Data Science and Analytics (DSA) Thrust, The Hong Kong University of Science and Technology (Guangzhou), Nansha, Guangzhou, Guangdong 511400, China, and also with The Hong Kong University of Science and Technology, Hong Kong, SAR, China (e-mail: leizhu@ust.hk).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TCSVT.2024.3424219.

Digital Object Identifier 10.1109/TCSVT.2024.3424219

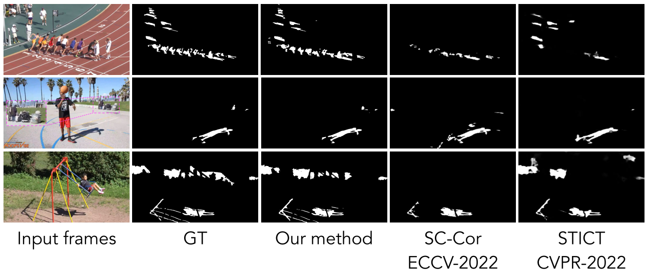


Fig. 1. Visual comparisons of shadow detection results produced by our method and two state-of-the-art methods in terms of video frames with multiple or tiny shadow regions. Apparently, our method can identify more shadow regions or more accurately detect tiny shadow regions than SC-Cor and STICT. GT denotes the ground truth of shadow detection on input video frames.

brings inconsistent predictions. Ding et al. [22] computed shadow-consistent correspondence to enhance pixel-wise similarity of the specific shadow regions across adjacent video frames and improve shadow detection performance. Although capturing the shadow objects in most scenarios, these methods tend to fail in detecting tiny shadow objects or handling video frames with multiple shadow objects. As shown in Figure 1, SC-Cor [22] tends to identify shadow regions with different region sizes, and their performance on detecting tiny shadow objects is degraded.

In this work, our objective is to address the challenge of identifying all shadow pixels of each video frame. Existing video shadow detection methods tend to fail in fully identifying all shadow regions, since they mainly learned temporal features at single-scale and single memory. Therefore, we develop a novel convolutional neural network (CNN) to learn motion-guided multi-scale memory features to obtain multi-scale temporal information based on multiple network memories for boosting video shadow detection. As illustrated in Fig. 1, we present visual comparisons of shadow detection results generated by our video shadow detection network and two leading and state-of-the-art methods, focusing on video frames with multiple or tiny shadow regions. It is evident that our method can identify a greater number of shadow regions or more precisely detect tiny shadow regions compared to SC-Cor and STICT.

Specifically, we devise a multi-scale motion-guided long-short transformer module (MMLT) to learn multi-scale temporal features and multi-scale motion features to detect shadows of the input video frame. In our MMLT module, we develop three dense-scale transformer blocks to learn multi-scale long-term features, multi-scale short-term features, and multi-scale motion features, and then integrate them to obtain a multi-scale transformer feature map, which is then passed into a decoder for shadow detection of the input video frame.

Moreover, we devise a self-supervised learning to predict an optical flow map from the input video frame for training the feature extraction encoder, the DST block and DLT block to compute multi-scale features.

Overall, our contributions are summarized as follows:

- We develop a novel convolutional neural network to learn motion-guided multi-scale temporal features for boosting video shadow detection by devising multi-scale motion-guided long-term transformer (MMLT) modules.
- In our MMLT module, we develop three dense-scale transformer blocks to learn a multi-scale long-term feature map, a multi-scale short-term feature map, and a multi-scale motion feature map, respectively. Our dense-scale transformer block contains a set of memory-read pooling attention (MPA) blocks and utilizes dense connections to integrate output features (with different scales) of MPA blocks to generate multi-scale output features.
- We devise a self-supervised task to predict the optical flow map from the input video frame for training the feature extraction encoder, the DST block, and DLT block for the subsequent multi-scale feature learning.
- Experimental results on three benchmark datasets show that our video shadow detection network clearly outperforms 26 state-of-the-art methods.

II. Related Work

A. Image Shadow Detection

Early works [10], [11], [12], [13], [26] mainly focused on exploring illumination models and color information, but these methods only worked well on high-quality images. Later, a number of shadow detectors [14], [15], [16], [17], [18], [19], [20], [26], [27], [28], [29] based on convolutional neural networks (CNNs) have been proposed to automatically identify shadow pixels of the input single image. Although these CNNs have achieved superior performance over classical shadow detectors, it is still not satisfactory to directly extend these CNNs trained on single images for video shadow detection due to a lack of learning temporal information among video frames.

B. Video Shadow Detection

Unlike single-image shadow detection, video shadow detection (VSD) aims to detect the shadow regions of each video frame. Early works [9], [30], [31] focus on utilizing hand-crafted features to detect the shadow regions of input videos. Recently, Chen et al. [8] collected a large-scale annotated video shadow detection dataset (ViSha) and presented a network with a dual gated co-attention module and a T-module to learn intra-video and inter-video features for video shadow detection. Afterward, Hu et al. [21] devised a warping module to align and combine features of neighboring video frames, while Lu et al. [23] presented an image-to-video shadow detection by utilizing the unlabeled video frames and labeled images. More recently, Ding et al. [22] learned shadow-consistent correspondence to enhance pixel-wise similarity of shadow regions across frames, Chen et al. [24] utilized existing labeled image dataset to produce pseudo-labels for semi-supervised video shadow detection. Liu et al. [32] highlight the importance of considering shadow deformation in video shadow detection methods. They propose two novel approaches: SODA, a self-attention module designed to handle large shadow deformations, and SCOTCH, a shadow contrastive learning mechanism that facilitates the learning of a unified shadow representation from

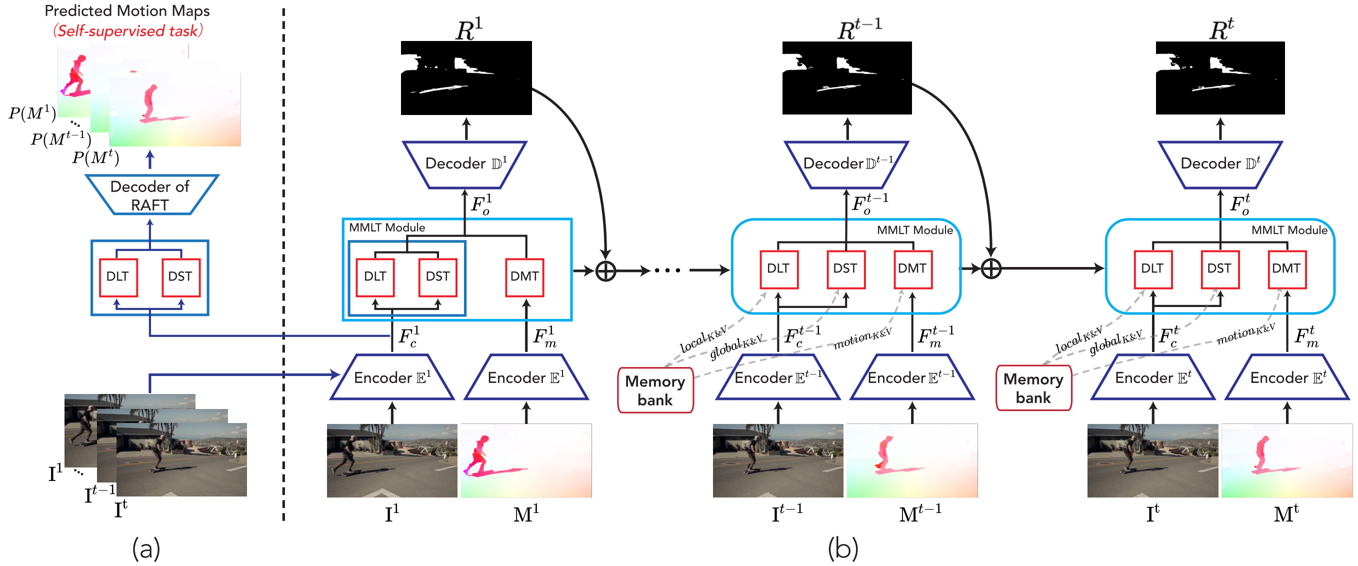


Fig. 2. The schematic illustration of our proposed MMLT-Net. Subfigure (a) illustrates the self-supervised pretraining process of our video shadow detection network. Specifically, it first passes the current video frame I^t and its adjacent frame I^{t-1} into an encoder identical to the video shadow detection encoder. Then, the encoder outputs are then feed into DLT and DST blocks to generate a feature map, which then passed into a RAFT decoder to predict an optical flow map $P(M^t)$. Subfigure (b) illustrates the training process for our video shadow detection model. We first pass I^t and M^t into two feature extraction encoders (denoted as $\mathbb{E}^t$) to obtain two feature maps (denoted as F_c^t and F_m^t). Then, we pass F_c^t and F_m^t into a MMLT module to obtain a multi-scale transformer features map F_o^t, which is then fed into a decoder $\mathbb{E}^t$ for predicting a shadow detection result of the input video frame. Note that the MMLT module denotes a motion-guided multi-scale long-short transformer module, and it consists of a dense-scale long transformer (DLT), dense-scale short transformer (DST), and dense-scale motion transformer (DMT); see Figure 3.

positive shadow pairs across multiple videos. Although these CNNs work well for many shadow videos, they fail to identify tiny shadows or multiple shadow regions due to limited temporal information learned from only several neighboring video frames.

C. Video Object Segmentation

Video object segmentation (VOS) automatically separates primary foreground objects from the background of each video frame [33], [34], [35], [36], [37]. Oh et al. [38] and its following works (e.g., EGMN [39] and Seong et al. [40]) leveraged a memory network to embed past-frame predictions into memory and applied a self-attention mechanism to read memory to decode the segmentation of the current frame. Duke et al. [41] utilized transformer blocks to extract pixel-level affinity maps and spatial-temporal features for video object segmentation. Yang et al. [42] addressed video object segmentation by associating objects with transformers.

D. Vision Transformers

Owing to their dominated performance for diverse natural language processing (NLP) tasks [43], [44], [45], many transformer networks were introduced to address many computer vision tasks, such as image classification [46], [47], [48], object detection [49], object segmentation [50], tracking [51], low-light video enhancement [52], and image generation [53], and these works also have shown promising performance over classical CNN-based methods. Liu et al. [48] introduced a shifted windowing scheme to limit self-attention computation to non-overlapping local windows and considered the cross-window interaction, while Carion et al. [49] first applied transformers into the object detection field to achieve end-to-end object detection manner. In this work, we leverage the transformer mechanism to learn multi-scale spatial-context features and multi-scale motion features to improve the VSD accuracy of detecting multiple shadow regions.

III. Proposed Method

A. Overview

Figure 2 shows the schematic illustration of our video shadow detection network. The intuition behind our network is to devise a motion-guided multi-scale long-term transformer (MMLT) module to learn multi-scale color-memory features and multi-scale motion-memory features for predicting the shadow mask of each video frame. To achieve this, we first construct a global color memory, a local color memory, and a motion memory from the first video frame, its corresponding motion map, and its shadow detection mask, and then iteratively update three memories by using intermediate feature maps (see p^t and q^t of Figure 3) of the MMLT module and the predicted shadow detection masks. Then, for the current video frame I^t, we compute its optical flow map M^t from I^t and I^{t-1} using RAFT [54], and then pass I^t and M^t into two feature extraction encoders ($\mathbb{E}^t$) to obtain two feature maps (denoted as F_c^t and F_m^t). Then, we pass F_c^t and F_m^t into a MMLT module to learn a multi-scale feature map F_o^t, which is then passed into a decoder $\mathbb{D}^t$ for predicting the shadow mask (R^t) of the input video frame I^t. Moreover, we devise a self-supervised task to learn an optical flow map $P(M^t)$ from the input I^t for pre-training the feature extraction encoder $\mathbb{E}^t$ in order to boost the downstream video shadow detection.

We utilize a FPN-like structure [55] to design our decoder. Specifically, we utilize skip connections to fuse highly semantic features representing global attributes of shadow regions and low-level but subtly delicate features. Then we apply a 3 × 3 convolutional layer and a 1 × 1 convolutional layer to predict the final shadow detection result R^t of the input video frame I^t.

B. Self-Supervised Optical Flow Prediction

Self-supervised learning has been utilized for many vision tasks by formulating diverse pre-text tasks, including predicting context [56] or image rotation [57]. The reason behind this is that the intermediate layers of convolutional neural networks (CNNs) trained for solving these pre-text tasks encode high-level semantic visual representations, which are capable of helping to address the downstream tasks of interest. Motivated by this, we devise a new self-supervised task, which predicts an optical flow map from the input video frame. By doing so, we can pre-train the feature extraction encoder, the DST block, and the DLT block for the downstream shadow detection of the current video frame.

As shown in Figure 2, we first pass the input video frame I^t and its adjacent video frame I^{t-1} into an encoder, which has the same structure as the encoder of the subsequent video shadow detection. Then we pass two output features of the encoder to a DLT block and a DST block to obtain a new feature map, which is then fed into a decoder of RAFT [54] for predicting an optical flow map $P(M^t)$. Here, we empirically remove the motion branch of the MMLT module since there is no motion map as the input for this self-supervised learning.

C. Multi-Scale Motion-Guided Long-Short Transformer (MMLT) Module

In the past few years, by storing and reading features of a number of video frames, memory networks have achieved superior performance on diverse applications, such as video object segmentation [38], [39], [42]. However, existing memory networks store and read features at a single scale, thereby degrading the performance of detecting multiple shadow regions in dynamic videos. To alleviate this issue, we develop a video shadow detection network to construct three memories and then devise a multi-scale motion-guided long-short transformer (MMLT) module to learn multi-scale transformer features from three memories for detecting shadows from the input video frame. Compared to existing memory networks, our method has two advantages: (1) Instead of only relying on spatial contexts, our method has three memories, which are a global memory from long-term video frames, a local memory from nearby short-term frames, and a motion memory from its motion map, to leverage object motion information for video shadow detection. (2) Our MMLT module has three dense-scale transformer modules to read features of three memories to learn three kinds of multi-scale memory features, and they are a multi-scale long-term feature, a multi-scale short-term feature, and a multi-scale motion feature. By doing so, our network can better identify multiple shadow regions.

1) Three Memory Construction and Updating: The classical STM [38] explores the idea of reading memory features to refine features of the current video frame for video object detection, and the memory features are often built from long-term memory frames. Later on, Yang et al. [42] explores two kinds of memory-based attention mechanisms for video object segmentation. They are a long-term attention from long-term memory frames and a short-term attention from nearby short-term frames. However, existing long-term memories and short-term memories are from only RGB video frame. In this work, we argue that the motion information provides a factor to detect video shadows, which tend to be dynamic in the input video. Motivated by this, we devise three memories to leverage color and motion information for reading memory features to refine features of the current video frame. Three memories include a global memory for storing features of long-term video frames, a local memory for storing features of short-term video frames, and a motion memory to store motion features.

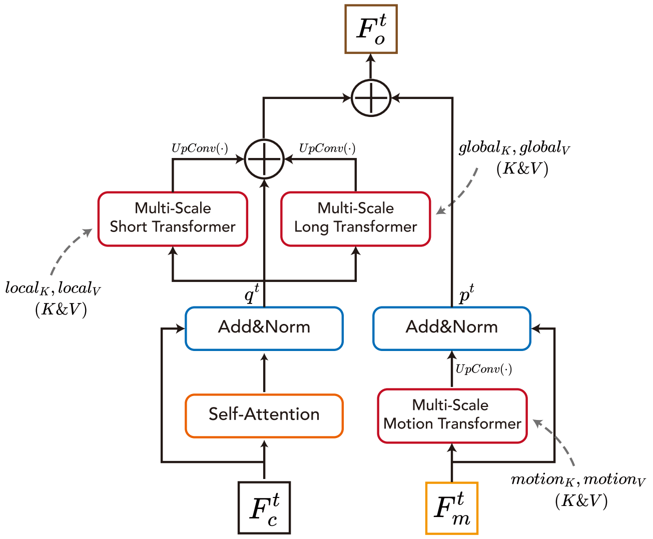

F_o^t
$UpConv(\cdot)$
$UpConv(\cdot)$
$global_K, global_V$ $(K\&V)$
Multi-Scale Short Transformer
Multi-Scale Long Transformer
$local_K, local_V$ $(K\&V)$
q^t
p^t
Add&Norm
Add&Norm
$UpConv(\cdot)$
Self-Attention
Multi-Scale Motion Transformer
$motion_K, motion_V$ $(K\&V)$
F_c^t
F_m^t

Fig. 3. The schematic illustration of our motion-guided multi-scale long-short transformer (MMLT) module, which consists of a dense-scale long transformer (DLT), dense-scale short transformer (DST), and dense-scale motion transformer (DMT).

Specifically, the local memory at t-th video frame has two features (denoted as $local_k^t$ and $local_v^t$) and the global memory also contains two features (denoted as $global_k^t$ and $global_v^t$), which are computed as:

$$\begin{aligned} local_k^t &= Cat_H(LT(q^{t-\delta+1}), \ldots, LT(q^{t-1}), LT(q^t)), \\ local_v^t &= Cat_H(LT(q^{t-\delta+1}), \ldots, LT(q^{t-1}) + LT(q^t) \\ &\quad + Conv(R^{t-1})), \\ global_k^t &= Cat_H(local_k^1, local_k^{1+\phi}, local_k^{1+2\phi}, \\ &\quad \ldots, local_k^{(1+\lfloor \frac{t}{\phi} \rfloor \phi)}), \\ global_v^t &= Cat_H(local_v^1, local_v^{1+\phi}, local_v^{1+2\phi}, \\ &\quad \ldots, local_v^{(1+\lfloor \frac{t}{\phi} \rfloor \phi)}), \end{aligned} \tag{1}$$

where $LT(\cdot)$ denotes a linear transformation layer (i.e., nn.Linear in Pytorch), and the parameters of different layers are not shared. $Cat_H(\cdot)$ denotes the feature concatenation operation along the horizontal direction of the 3D feature maps. R^{t-1} represents the shadow detection result of the $(t-1)$-th video frame. For the first frame, we empirically utilize a single shadow detection method (i.e., BDRAR [18]) to predict a shadow detection mask as R^{t-1}. $Conv(\cdot)$ is a 3×3 convolutional layer. $\lfloor . \rfloor$ denotes an integer floor function. δ represents the number of previous video frames stored in the local memory of our method. ϕ denotes the intervals between video frames for updating the global memory in our method. Note that we update a global memory after each ϕ frames, while the local memory has previous δ video frames of the current video frame. And the motion memory has two features (denoted as $motion_k^t$ and $motion_v^t$) at t-th video frame is updated as:

$$\begin{aligned} motion_k^t &= LT(p^t), \\ motion_v^t &= FC(p^t) + Conv(R^{t-1}), \end{aligned} \tag{2}$$

where $LT(\cdot)$ denotes a linear transformation layer and two linear transformation layers in $motion_k^t$ and $motion_v^t$ do not share the learning parameters. R^{t-1} represents the shadow detection result of the $(t-1)$-th video frame; see Eq. (1).

2) MMLT Module: With these three memories, we devise a multi-scale motion-guided long-short transformer (MMLT) module to learn motion-guided multi-scale features for shadow detection of I^t from F_c^t and F_m^t, which are two features obtained by passing the input video frame I^t and its optical flow map M^t into the encoder.

Fig. 3 shows the schematic illustration of our MMLT module. Specifically, our MMLT module first passes the color feature F_c^t into a self-attention module [58] to obtain a new feature map, which is then element-wisely added with F_c^t. We then apply a layer optimization to produce a feature map q^t. Then, we pass the motion feature F_m^t into a multi-scale transformer (named multi-scale motion transformer) with the motion memory and then pass the output feature into a $UpConv(\cdot)$ operation to obtain a feature map p^t. The $UpConv(\cdot)$ operation contains an upsampling operation and a 3×3 convolutional layer. The upsampling operation is used to rescale the feature map size to the same size as the input feature map (i.e., F_c^t or F_m^t) of the MMLT module, while the 3×3 convolutional layer is to ease the effect of feature aliasing due to the upsampling operation.

After that, we pass q^t into a multi-scale short-term transformer with the local memory, and a multi-scale long-term transformer with the global memory to obtain two features. We then pass the two obtained features into two $UpConv(\cdot)$ operations to obtain a new feature map, which is then element-wisely added with p^t to generate the output feature F_o^t of our MMLT module. Mathematically, the output F_o^t of our MMLT module at the t-th video frame can be computed by:

$$\begin{aligned} F_o^t &= UpConv(DLT(p^t)) + UpConv(DST(p^t)) + p^t, \\ p^t &= UpConv(DMT(F_m^t)) + q^t, \\ q^t &= SelfAttn(F_c^t) + F_c^t. \end{aligned} \tag{3}$$

where $SelfAttn(\cdot)$ denotes a self-attention module; see [58] for details. "DLT", "DST", and "DMT" represent the multi-scale long-term transformer, multi-scale short-term transformer, and multi-scale motion transformer, respectively.

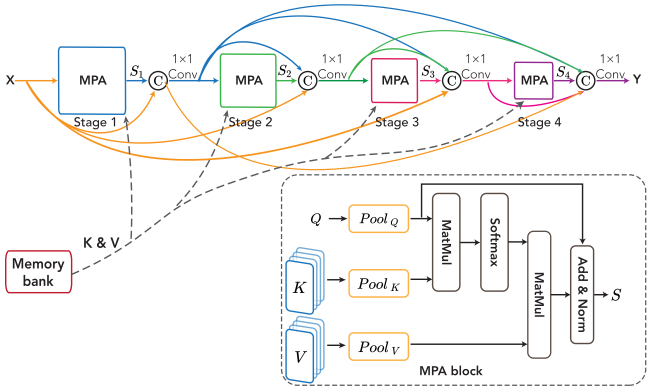


Fig. 4. The schematic illustration of our dense-scale transformer block, which is to build DST, DLT, and DMT of our MMLT module. MPA block denotes the memory-read pooling attention block.

Three $UpConv(\cdot)$ operations do not share the same parameters.

3) Dense-Scale Transformer: Note that our MMLT module in Figure 3 has three dense multi-scale transformers, including a dense-scale motion transformer (DMT), dense-scale short transformer (DST), and dense-scale long transformer (DLT). These three transformers are based on our dense-scale transformer, but they have different input feature maps and different memory features. Unlike the original multi-scale vision transformers [59], our dense-scale transformer block densely connects the output features of our memory-read pooling attention (MPA) block at different stages to promote feature integration for learning multi-scale transformer features. By doing so, we can better detect shadows with different region sizes, thereby boosting video shadow detection of our method. Figure 4 shows the schematic illustration of our dense-scale transformer. Apparently, our dense-scale transformer has four stages, and each stage contains a memory-read pooling attention (MPA) block. The output feature map of the MPA block at different stages has specific spatial and channel dimensions. Specifically, the output features (denoted as S_1, S_2, S_3, and S_4) of all four MPA blocks are computed by:

$$\begin{aligned} S_1 &= MPA(X, K, V)), \\ S_2 &= MPA(Conv(Cat(X, S_1)), K, V))), \\ S_3 &= MPA(Conv(Cat(X, S_1, S_2)), K, V))), \\ S_4 &= MPA(Conv(Cat(X, S_1, S_2, S_3)), K, V))), \end{aligned} \tag{4}$$

where MPA denotes the memory-read pooling attention (MPA) block, which takes three features as the inputs, and outputs a fused feature map. $Conv(\cdot)$ represents a 1×1 convolutional layer. $Cat(\cdot)$ is the feature concatenation operation along the feature channel direction. After that, the output feature (denoted as Y) of our dense-scale transformer block is computed by:

$$Y = Conv(Cat(X, S_1, S_2, S_3, S_4)). \tag{5}$$

4) Memory-Read Pooling Attention (MPA) Block: Note that the channel dimension and the spatial resolution in self-attention blocks are often fixed for the output feature map. To change the spatial resolution, multi-head pooling attention [59] has been developed to refine an input feature map by adding a pooling attention mechanism into a self-attention block. Motivated by the superior performance of the multi-head pooling attention, we devise a memory-read pooling attention (MPA) block to read the memory features to refine features of the current video frame to enhance video shadow detection performance. Figure 4 shows the schematic illustration of our MPA block. It reads two memory features (denoted as K and V) from the memory to refine the input feature (denoted as Q). For the current t-th video frame, we compute two memory features (denoted as K and V) by concatenating the memory features from the first video frame to the $(t-1)$-th video frame.:

$$\begin{aligned} K &= Cat(LT(\alpha^1), LT(\alpha^2), \ldots, LT(\alpha^i), \ldots, \alpha^{t-1}), \\ V &= Cat(LT(\alpha^1), LT(\alpha^2), \ldots, LT(\alpha^i), \ldots, \alpha^{t-1})), \end{aligned} \tag{6}$$

where $Cat(\cdot)$ denotes the feature concatenation operation along the feature channel direction, while $LT(\cdot)$ represents the linear transformation layer. α^i is q^t at Eq. 1 for the MPA block in the dense-scale short transformer or dense-scale long transformer, while α^i is p^t at Eq. 2 for the MPA block in the dense-scale motion transformer. Moreover, the input feature maps (denoted as Q_1, Q_2, Q_3, and Q_4) Q at the four stages are computed by:

$$\begin{aligned} Q_1 &= X, \\ Q_2 &= Cat(X, S_1), \\ Q_3 &= Cat(X, S_1, S_2), \\ Q_4 &= Cat(X, S_1, S_2, S_3), \end{aligned} \tag{7}$$

where S_1, S_2, S_3, S_4 denote the output feature map of our MPA block at the four stages.

Once obtaining three feature maps (Q, K, V), our MPA block first reshapes them into 2D matrices, and then applies three pooling operations to them to reduce the spatial resolution by half. After that, we follow the classical self-attention mechanism to multiply the reshaped matrices on Q and K to compute a similarity matrix, which is then passed into a $Softmax(\cdot)$ to normalize the similarity matrix. Then, we multiply the normalized similarity matrix with the reshaped V and add the multiplication result with the reshaped Q to produce the output feature (S) of our MPA. Mathematically, the definition of S is given by:

$$S = \mathcal{P}(Q;\theta) + Softmax\left(\mathcal{P}(Q;\theta)\mathcal{P}(K;\theta)^T\right)\mathcal{P}(V;\theta), \tag{8}$$

where $\mathcal{P}(:;\theta)$ denote a pooling operation on a feature map along three dimensions with a kernel $\theta = (\theta_k, \theta_s, \theta_p)$. $\theta_k = (\theta_k^h, \theta_k^w)$, $\theta_s = (\theta_s^h, \theta_s^w)$, and $\theta_p = (\theta_p^h, \theta_p^w)$ represent the kernel, stride, and padding of the pooling operation. In MPA blocks of all stages, θ_k, θ_s, and θ_p are empirically set as $\theta_k = (3, 3)$, $\theta_s = (2, 2)$, and $\theta_p = (1, 1)$ in our experiments.

5) Three Dense-Scale Transformer Modules (DMT, DLT, DST): As shown in Figure 4, the dense-scale transformer blocks have three inputs, and they are the input feature X, and two memory features (see K and V of Figure 4). And we utilize this dense-scale transformer block to build DMT, DLT, and DST of our MMLT module, and the differences of DMT, DLT, and DST are summarized as follows:

- **Dense-scale motion transformer (DMT).** DMT exploits the video motion information to emphasize potential shadow regions of the current video frame. It leverages the motion memory features to refine the motion feature map from the current video frame for shadow detection. Hence, three input features of DMT include the motion feature map (F_m^t) and two motion memory features $(motion_k$ and $motion_v)$.
- **Dense-scale long transformer (DLT).** DLT aggregates shadow information from the long-term RGB memory to refine features of the current video frame for shadow detection. Hence, the three input features of DLT include the RGB feature map (F_c^t) and two long-term memory features $(global_k$ and $global_v)$.
- **Dense-scale short transformer (DST).** learns a temporal information from nearby short-term video frames for detecting shadows from the current video frame. Hence, the three input features of DLT include the RGB feature map (F_c^t) and two local memory features $(local_k$ and $local_v)$.

D. Loss Function

Our network has two training stages. The first stage is to predict an optical flow map $P(M^t)$ from the input video frame I^t, and we utilize a L_1 loss to compute the optical flow prediction error:

$$\mathcal{L}_{optical} = \Omega_{L1}(\mathbb{M}_p, \mathbb{M}_{gt}) \tag{9}$$

where Ω_{L1} is the L_1 loss function. $\mathbb{M}_p$ (see $P(M^t)$ of Figure 2) and $\mathbb{M}_{gt}$ denote the predicted optical flow map and the ground truth of the optical flow, respectively.

In the second training stage, we predict a shadow detection map R^t of the input video frame I^t. Then, we compute a shadow detection error L_{seg} by using a binary cross-entropy (BCE) loss and an IoU [17] loss:

$$\mathcal{L}_{seg} = \lambda_1\Omega_{BCE}(\mathbb{R}_p, \mathbb{R}_{gt}) + \lambda_2\Omega_{IoU}(\mathbb{R}_p, \mathbb{R}_{gt}) \tag{10}$$

where $\mathbb{R}_p$ and $\mathbb{R}_{gt}$ denote the predicted shadow detection map and the underlying ground truth. Ω_{BCE} and Ω_{IoU} are the binary cross-entropy (BCE) loss and an IoU loss, respectively. λ_1 and λ_2 are to balance the BCE loss and the IoU loss, and we empirically set them as $\lambda_1 = 0.5$, $\lambda_2 = 0.5$ in our experiments.

IV. Experiments

A. Datasets and Evaluation Metrics

We evaluate the effectiveness of the proposed video shadow detection (VSD) method on three challenging benchmark datasets (i.e., ViSha [8], VISAD-DS [23] and VISAD-MOS [23]). ViSha [8] is a significant milestone as it is the first large-scale dataset for video shadow detection. It comprises 120 videos with 11,685 image frames. Moreover, VISAD-DS [23] and VISAD-MOS [23] are two subparts of VISAD [23], each containing Driving Scenes (DS) and

TABLE I

QUANTITATIVE COMPARISONS BETWEEN OUR VIDEO SHADOW DETECTION NETWORK AND THE STATE-OF-THE-ART METHODS IN TERMS OF MAE, F_β, IOU, AND BER ON VISHA DATASET

Method	MAE ↓	F_β ↑	IoU ↑	BER ↓
BDRAR [18]	0.050	0.695	0.484	21.29
DSD [20]	0.044	0.702	0.518	19.88
MTMT [27]	0.043	0.729	0.517	20.28
FPN [55]	0.044	0.707	0.512	19.49
PSPNet [60]	0.051	0.642	0.476	19.75
DSS [61]	0.045	0.696	0.502	19.77
R^3Net [62]	0.043	0.710	0.502	20.40
PDBM [63]	0.066	0.623	0.466	19.73
COSNet [33]	0.040	0.705	0.514	20.50
MGA [64]	0.067	0.601	0.399	25.77
FEELVOS [65]	0.043	0.710	0.512	19.76
STM [38]	0.068	0.597	0.408	25.69
TVSD-Net [8]	0.033	0.757	0.567	17.70
TFW [21]	0.078	0.683	0.510	17.03
STICT [23]	0.046	0.702	0.545	16.60
SC-Cor [22]	0.042	0.762	0.615	13.61
MPLNet [24]	-	-	-	8.69
SCOTCH and SODA [32]	0.029	0.793	0.640	9.07
our method	**0.019**	**0.817**	**0.741**	**7.2**

Moving Object Scenes (MOS), respectively. To quantitatively compare different VSD methods, we employ four evaluation metrics, namely, Mean Absolute Error (MAE), F-measure (F_β), Intersection over Union (IoU), and Balanced Error Rate (BER). MAE measures the average pixel-level relative error between the ground truth and the predicted result by calculating the mean of the absolute value of their differences. F_β provides a comprehensive assessment of both precision and recall, resulting in a weighted harmonic mean given by: $F_\beta = (1+\beta^2)\frac{P*R}{\beta^2 P+R}$, where β^2 is set to 0.3 to emphasize the precision score. IoU describes the extent of overlap between the predicted results and the ground-truth map. BER is the average of errors for each class, calculated as follows: $BER = (1 - \frac{1}{2}(\frac{TP}{N_p} + \frac{TN}{N_n})) \times 100$, where TP and TN denote the true positives and the true negatives, respectively. N_p and N_n represent the number of shadow pixels and the non-shadow pixels, respectively. In general, a better video shadow detector often has smaller BER and MAE scores as well as larger F_β and IoU scores.

B. Implementation Details

We implement our MMLT-Net using PyTorch, and train it by using an Adam optimizer on four NVIDIA GTX 2080Ti. We initialize the feature extraction backbone via a MobileNet-V2 [70] pre-trained using our self-supervised task of predicting an optical flow map, while other parameters are trained from scratch. The model needs to take about 20 hours to converge. We resize the input RGB image and its optical flow image to 448 × 448 and adopt three data augmentation operations, including random flipping, rotating, and border clipping. The weight decay, batch size, and iteration number are empirically set as 0.0005, 4, and 20000, respectively. We set the initial learning rate as 0.0004 for scratch layers and 0.00005 for pre-trained layers and then used a cosine decay with a warm-up period to adjust the learning rate. During the testing stage, each video frame and its optical flow map are resized to 448 × 448 and then fed into our network to predict a shadow detection map of the input video frame. Finally, a bilinear interpolation is applied to up-sample the prediction map to the original size of the input video frame to achieve its final shadow detection.

C. Performance Comparison

1) Compared Methods: We compare our method against 26 state-of-the-art VSD methods, including BDRAR [18], DSD [20], MTMT [27], FPN [55], PSPNet [60], DSS [61], R^3Net [62], PDBM [63], COSNet [33], MGA [64], STM [38], FEELVOS [65], DSC [17], ECANet [66], FSDNet [28], MagNet [67], SANet [23], GRF [21], NS [68], RCRNet [69], TVSD-Net [8], Hu et al. [21], STICI [23], SC-Cor [22], MPLNet [24], as well as SCOTCH and SODA [32].

2) Quantitative Comparisons: Table I and Table II summarize the quantitative results of our method and state-of-the-art video shadow detection methods in terms of all four metrics, including MAE, F_β, IoU, and BER on three benchmark datasets. From the quantitative results on the ViSha dataset, we can find that TVSD-Net [8] has the smallest MAE score of 0.033; SC-Cor [22] has the largest F_β score of 0.762 and the largest IoU score of 0.615; while MPLNet [24] has the smallest BER score of 8.69 among all compared methods. (Note that MPLNet [24] does not release their MAE, F_β, and IoU scores). Compared to these best-performing existing methods, our network obtains a MAE improvement of 42.4%, a F_β improvement of 7.2%, an IoU improvement of 20.5%, and a BER improvement of 17.1%, respectively. Specifically, the MAE, F_β, IoU, and BER scores of our network are 0.019, 0.817, 0.741, and 7.2.

Regarding VIDAD-DS and VISAD-MOS, our network has better metric results than compared methods for all four metrics. Specifically, our method has a MAE score of 0.027, a F_β score of 0.732, a IoU score of 0.533, and a BER score of 9.23 for the VISAD-DS dataset. And the MAE, F_β, IoU, and BER scores of our method on the VISAD-MOS dataset are 0.051, 0.639, 0.441, and 15.32, respectively.

3) Visual Comparisons: Figure 5 visually compares video shadow detection results of our network and state-of-the-art methods for input video frames, which contain multiple shadow regions. From the visual results, we can find that compared methods tend to miss parts of shadow regions or wrongly identify non-shadow regions as target ones, especially for small shadow regions. On the contrary, our method (see 3rd column of Figure 5) can more accurately identify all shadow regions of input video frames.

D. Ablation Study

1) Baseline Design: We construct seven baselines to evaluate the effectiveness of three dense-scale transformer blocks, including the dense-scale short transformer (DST), the dense-scale long transformer (DLT), and the dense-scale motion transformer (DMT). The first baseline network (denoted as "basic") is reconstructed by removing the motion branch and the self-supervised task from our network, and

TABLE II

QUANTITATIVE COMPARISONS BETWEEN OUR VIDEO SHADOW DETECTION NETWORK AND THE STATE-OF-THE-ART METHODS IN TERMS OF MAE, F_β, IOU, AND BER ON VISAD-DS AND VISAD-MOS DATASETS

Method	VISAD-DS				VISAD-MOS			
	MAE ↓	F_β ↑	IoU ↑	BER ↓	MAE ↓	F_β ↑	IoU ↑	BER ↓
DSC [17]	0.096	0.507	0.315	18.24	0.070	0.573	0.385	24.18
BDRAR [18]	0.088	0.504	0.284	15.25	0.130	0.456	0.250	18.79
DSD [20]	0.068	0.408	0.301	18.42	0.083	0.595	0.365	19.62
MTMT-SSL [27]	0.106	0.521	0.298	19.49	0.085	0.575	0.402	25.61
ECANet [66]	0.037	0.583	0.379	23.67	0.078	0.565	0.336	28.68
FSDNet [28]	0.029	0.623	0.377	23.67	0.078	0.565	0.336	28.68
MagNet [67]	0.038	0.606	0.399	21.56	0.080	0.586	0.341	28.91
SANet [23]	0.028	0.708	0.514	13.14	0.091	0.601	0.341	25.93
MTMT-Uns. [27]	0.154	0.309	0.232	20.19	0.081	0.564	0.391	27.01
TVSD-Net [8]	0.032	0.634	0.508	11.55	0.191	0.313	0.227	20.24
GRF [21]	0.057	0.611	0.326	18.87	0.115	0.551	0.292	26.76
NS [68]	0.073	0.495	0.339	19.16	0.115	0.534	0.261	29.60
RCRNet [69]	0.067	0.377	0.236	28.76	0.088	0.596	0.356	20.13
STICT [23]	0.065	0.646	0.370	14.17	0.058	0.625	0.409	18.51
SC-Cor [22]	0.055	0.653	0.393	13.01	0.053	0.632	0.424	16.51
Ours	**0.027**	**0.732**	**0.533**	**9.23**	**0.051**	**0.639**	**0.441**	**15.32**

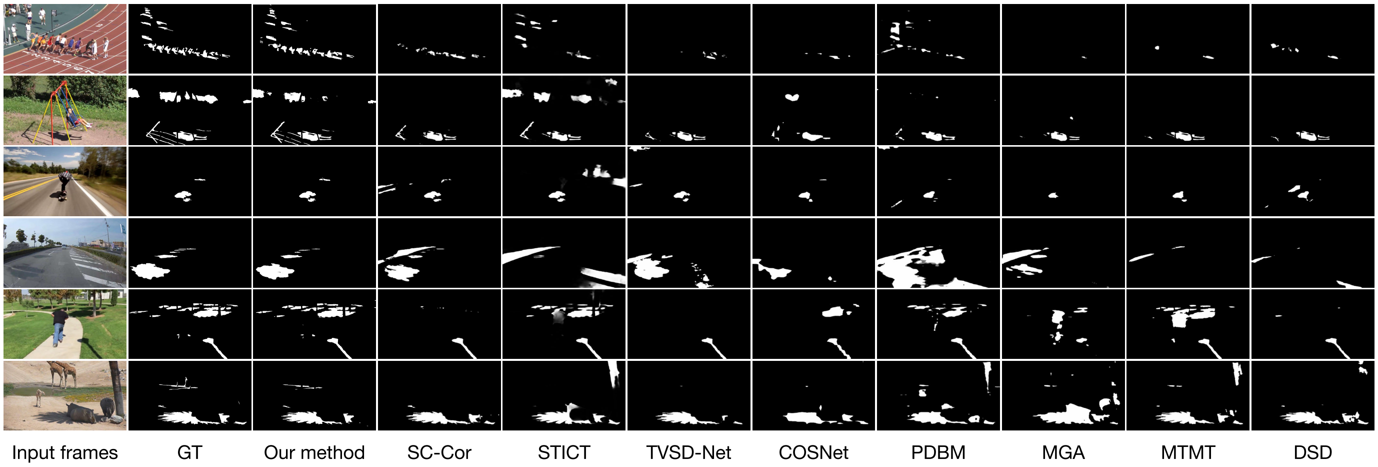


Fig. 5. Visual comparisons of video shadow detection results produced by our network (3rd column; denotes as "Ours") and state-of-the-art methods (4th to 11-th columns) against ground truths (2nd column). Apparently, our network can more accurately identify shadow regions than all compared methods.

then utilizing the original long-term attention and short-term attention of [42] to replace DLT and DST of our network. The second baseline network (denoted as"basic+DST") is to replace the original short-term attention [42] by using DST of our framework. Similarly, we can obtain the third (denoted as "basic+DLT") and the fourth (denoted as "basic+DMT") baseline networks by using DLT or DMT within our framework. The fifth baseline network (denoted as "basic+DST+DLT") is built by replacing the original long-term attention of "basic+DST" by using our DLT. The sixth baseline network (denoted as "basic+DLT+DMT") is created by adding our DMT to "basic+DLT". The eighth baseline network (denoted as "basic+DST+DLT+DMT") is to add our DMT into "basic+DST+DLT". Apparently, "basic+DST+DLT+DMT" includes all three dense-scale transformer blocks (i.e., DST, DLT, and DMT). It is equal to removing the self-supervised task of predicting the optical flow from our network. And the last baseline network (denoted as "basic+Conv") is constructed by replacing these three modules with many convolutional layers, which have the same number of parameters as these three modules.

2) *Quantitative Comparisons:* Table III reports the quantitative results of our method and four baseline networks in terms of MAE, F_β, IoU, and BER. Apparently, we can find that "basic+DST","basic+DLT", and "basic+DMT" outperforms "basic" for all four metrics, which demonstrates that incorporating the DST, DLT, or DMT modules can enhance video shadow detection performance. Then, "basic+DST+DLT" has smaller MAE and BER scores as well as larger F_β and IoU scores than "basic+DST". It indicates that the dense-scale long-term transformer (DLT) enables our network to better identify shadows from video frames. Similarly, "basic+DLT+DMT" has a superior performance over "basic+DLT" and "basic+DMT". It demonstrates that combining the DLT and the DMT together can further improve our video shadow detection performance. Moreover, "basic+DST+DLT+DMT" has a superior metric performance over "basic+DST+DLT". It means that the motion information from our dense-scale motion transformer (DMT) can help our network to better detect shadow pixels of video frames. The better metric results our method over "basic+DST+DLT+DMT" demonstrates that utilizing

TABLE III

QUANTITATIVE RESULTS OF OUR METHOD AND CONSTRUCTED BASELINE NETWORKS OF OUR METHOD IN TERMS OF MAE, F_β, IOU, AND BER. DMT, DLT, AND DST DENOTE THE DENSE-SCALE MOTION TRANSFORMER, THE DENSE-SCALE LONG TRANSFORMER, THE DENSE-SCALE SHORT TRANSFORMER, RESPECTIVELY. AND "SS" DENOTES THAT WE UTILIZE AN OPTICAL FLOW PREDICTION AS A SELF-SUPERVISED PRE-TRAINING TASK

Method	DST	DLT	DMT	SS	MAE ↓	F_β ↑	IoU ↑	BER ↓	#Param.(M)	FPS
basic	×	×	×	×	0.027	0.770	0.673	10.5	**5**	**25**
basic+DST	✓	×	×	×	0.023	0.796	0.685	11.4	37	22
basic+DLT	×	✓	×	×	0.025	0.790	0.679	11.5	37	22
basic+DMT	×	×	✓	×	0.024	0.792	0.683	9.2	38	21
basic+DST+DLT	✓	✓	×	×	0.019	0.808	0.717	8.3	59	20
basic+DLT+DMT	×	✓	✓	×	0.022	0.802	0.710	8.9	60	19
basic+DST+DLT+DMT	✓	✓	✓	×	0.021	0.798	0.723	7.8	82	18
Our method	✓	✓	✓	✓	**0.019**	**0.817**	**0.741**	**7.2**	82	18
basic+Conv	×	×	×	×	0.024	0.795	0.686	11.5	82	22

TABLE IV

ABLATION ANALYSIS RESULTS OF OUR METHOD WITH AND WITHOUT DENSE CONNECTION IN MMLT

Method	MAE ↓	F_β ↑	IoU ↑	BER ↓	FPS	FLOPs(G)	#Param. (M)
Our method	**0.019**	**0.817**	**0.741**	**7.2**	18	163	82
our-w/o-DC	0.022	0.786	0.733	7.8	**22**	**144**	**47**
TVSD-Net	0.033	0.757	0.567	17.70	12	159	62
SCOTCH and SODA	0.029	0.793	0.640	9.07	9	174	91

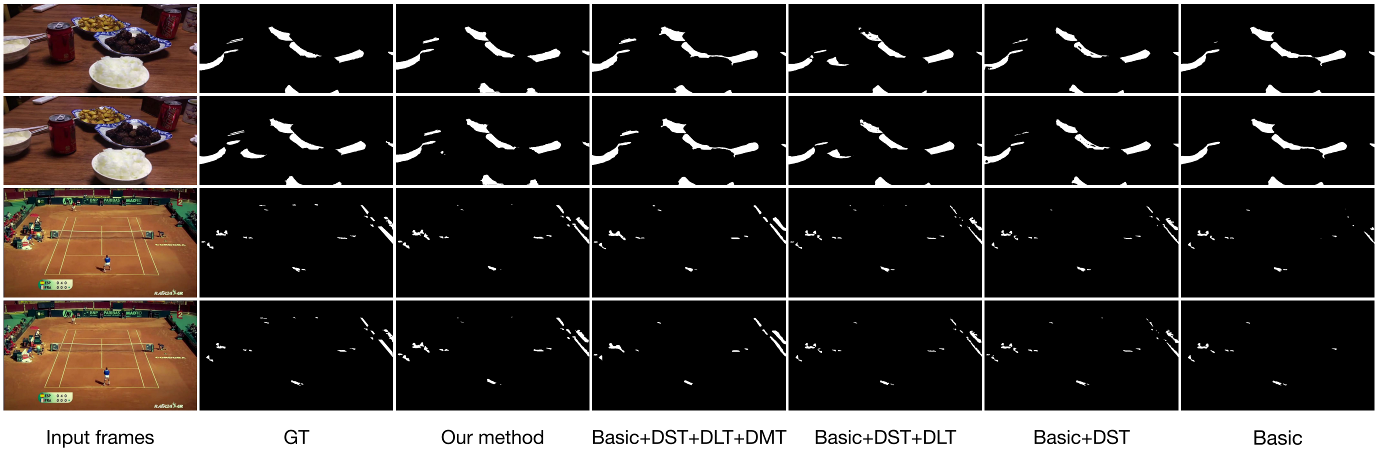


Fig. 6. Visual comparisons of video shadow detection results produced by our method and constructed baseline networks of our ablation studies.

the self-supervised task of an optical flow prediction to pre-training the feature extraction encoder can boost the downstream video shadow detection of our network. Note that Table III also summarizes the computation time and numbers of parameters (denotes as #Param.) of baseline networks of our ablation study to evaluate the speed and size of our three main modules. Apparently, the DST reduces the FPS from 25 to 22; DLT can further reduce the speed from 22 FPS to 20 FPS, and DMT reduces the FPS from 20 to 18. Since the difference between our method and "basic+DST+DLT+DMT" is the self-supervised learning, and the inference stage does not involve the self-supervised task, our method has the same FPS and the numbers of parameters as "basic+DST+DLT+DMT". To further verify whether our network performance comes from our three modules (i.e., DLT, DST, and DMT) or the increase of network parameters, we build another baseline network (denoted as "basic+Conv") by replacing these three modules with many convolutional layers, which have the same number of parameters as these three modules. By comparing "basic+Conv" and our network, we can find that our network has a better video shadow detection performance than "basic+Conv". It indicates that our network performance comes from the effectiveness of our three modules (DLT, DST, and DMT).

TABLE V

ABLATION ANALYSIS OF THE SELF-SUPERVISED LEARNING MODULE

Method	MMLT	MAE	F_β	IoU	BER
Ours-ss-before-MMLT	before	0.021	0.804	0.726	7.8
Our method	after	0.019	0.817	0.741	7.2

3) Visual Comparisons: In Figure 6, we present a visual comparison of detection results. It is evident that our network excels in accurately identifying shadow pixels in video frames compared to all five baseline networks. This observation further underscores the effectiveness of the three transformer blocks and the self-supervised task incorporated into our network.

TABLE VI

ABLATION ANALYSIS RESULTS OF OUR METHOD WITH DIFFERENT NUMBER OF PREVIOUS VIDEO FRAMES IN THE LOCAL MEMORY OF OUR METHOD

δ	MAE ↓	F_β ↑	IoU ↑	BER ↓
1 (ours)	**0.019**	**0.817**	**0.741**	**7.2**
2	0.019	0.807	0.739	8.0
3	0.018	0.814	0.735	8.0

4) Module for Self-Supervised Learning: We investigated the impact of self-supervised learning on various modules. We constructed and compared a baseline called 'ours-ss-before-MMLT', where self-supervised learning was applied only to the encoder. As shown in Tab. V our network outperforms 'ours-ss-before-MMLT' for all the four quantitative metrics. It indicates that the self-supervised learning after the MMLT in our work can more accurately detect shadows than that before the MMLT. Hence, the pre-training DST and DLT blocks via the self-supervised learning enable our network to produce more accurate shadow detection results.

5) The Effectiveness of Dense Connection in MMLT: Table IV summarizes MAE, F_β, IoU, and BER scores of our method and that (denoted as "ours-w/o-DC") without dense connection in MMLT. Obviously, our method has better MAE, F_β, IoU, and BER results than "ours-w/o-DC" and the other two SOTA methods. The reason behind is that the dense connection can fuse highly semantic features representing global attributes of shadow regions and low-level but subtly delicate features. Note that incorporating dense connections within our method would increase the complexity in terms of both time and space. Our approach results in a reduction of FPS from 22 to 18, while also increasing FLOPs(G) from 144 to 163 and the number of parameters in the model from 47 to 82. We also compare the complexity and inference speed between our network and two SOTA methods. Apparently, our FLOPs and the number of parameters are larger than TVSD-Net, but smaller than the more recent method "SCOTCH and SODA". However, our method is faster than TVSD-Net (12 FPS) and "SCOTCH and SODA" (9 FPS) due to our larger FPS score (18). Moreover, our network outperforms these two methods in terms of video shadow detection performance.

6) The Number of Previous Video Frames of the Local Memory: We utilize the features of previous video frames to compute the local memory in our approach. As a result, we conduct an ablation study experiment to compare the results of our method using different numbers of previous video frames for updating the local memory. Table VI presents a summary of the MAE, F_β, IoU, and BER scores of our method when using 1, 2, and 3 previous video frames in the local memory. Evidently, our method, which involves taking only the last video frame to update the local memory, demonstrates the best metric performance. Introducing more previous video frames would result in increased feature misalignment among video frames during the updating process of the local memory, thereby leading to a degradation in VSD performance.

7) The Number of Video Frame Intervals in Our Global Memory: Note that the global memory stores the feature maps of previous video frames with a fixed interval. Here, we conduct an ablation study experiment to study the performance with different video frame intervals. Table VII shows the VSD results of our method with the video frame interval of 3, 5, 7, and 9. Apparently, our method with 5-frame interval for updating the global memory has the best performance of MAE, F_β, IoU, and BER. Hence, our method empirically sets the video frame interval of 5 to update the global memory for video shadow detection.

TABLE VII

ABLATION ANALYSIS RESULTS OF OUR METHOD WITH DIFFERENT VIDEO FRAME INTERVALS OF UPDATING THE GLOBAL MEMORY OF OUR METHOD

ϕ	MAE ↓	F_β ↑	IoU ↑	BER ↓
3	0.018	0.812	0.733	8.9
5 (ours)	**0.019**	**0.817**	**0.741**	**7.2**
7	0.019	0.809	0.740	8.1
9	0.020	0.811	0.737	8.4

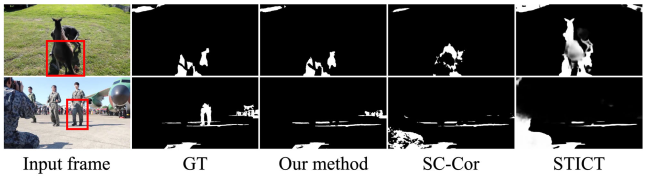


Fig. 7. Comparison of visual failure cases analysis.

TABLE VIII

ANALYSIS OF TEMPORAL CONSISTENCY FOR OUR METHOD AND OTHER STATE-OF-THE-ART METHODS

Method	TVSD	STICT	SC-Cor	Ours
E_{warp}	344.13	427.76	426.77	164.71

E. Failure Cases

Our model struggles to detect shadow regions with relatively low-contrast boundaries. Fig. 7 shows some failure cases of our method. As depicted in the first row of the figure below, like state-of-the-art methods, our network also fails in identifying weak shadows on kangaroos' bodies and human legs.

F. Temporal Consistency

Here, we introduce a widely-used flow warping error (E_{warp}) [71] to measure the temporal consistency of different video shadow detection results. Specifically, following [54], we follow [71] to utilize RAFT [54] to obtain the optical flow maps for computing E_{warp}. Table VIII reports E_{warp} scores video shadow detection results of our network and three state-of-the-art (SOTA) methods. Apparently, our network has the smallest E_{warp} score among all methods. It indicates that the shadow detection outcomes produced by our network demonstrate a superior temporal stability over all compared SOTA methods.

V. CONCLUSION

This paper presents a novel network for boosting video shadow detection (VSD) by learning motion-guided multi-scale memory features. The main idea of our network is to devise three dense-scale transformer blocks to learn multi-scale spatial context and memory features from three memory, thereby improving VSD performance on detecting multiple shadow regions. Moreover, the dense-scale transformer leverages a set of memory-read pooling attention (MPA) blocks to read memories for refining features and utilize dense connections to fuse output features of MPA with different scales. Experimental results on three benchmark datasets show that our network quantitatively and qualitatively outperforms state-of-the-art methods.

VI. Future Work and Social Impact

In our forthcoming research, we aim to enhance the model's ability to detect shadows in low light conditions, which often display distinct characteristics including soft edges, low visibility, and low contrast. To this end, we intend to compile a comprehensive shadow detection dataset under such conditions and develop a novel model that surpasses the performance of current state-of-the-art techniques. Regarding the potential social impacts of our research, we envision that improved shadow detection can have broad applications across various domains, including autonomous driving, AR/VR, data annotation, interactive image editing, etc. In the context of autonomous driving, shadows are sometimes erroneously recognized as tangible objects, resulting in flawed decisions. These misjudgements, under certain circumstances, could significantly endanger driver safety. Thus, enhancing autonomous driving technologies to accurately differentiate between shadows and real objects is essential for boosting driving safety.

References

[1] T. Okabe, I. Sato, and Y. Sato, "Attached shadow coding: Estimating surface normals from shadows under unknown reflectance and lighting conditions," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep. 2009, pp. 1693–1700.

[2] K. Karsch, V. Hedau, D. Forsyth, and D. Hoiem, "Rendering synthetic objects into legacy photographs," *ACM Trans. Graph.*, vol. 30, no. 6, pp. 1–12, Dec. 2011.

[3] N. Inoue and T. Yamasaki, "Learning from synthetic shadows for shadow detection and removal," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 11, pp. 4187–4197, Nov. 2021.

[4] L. Jie and H. Zhang, "RMLANet: Random multi-level attention network for shadow detection and removal," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 12, pp. 7819–7831, Dec. 2023.

[5] W. Wu, W. Yang, W. Ma, and X.-D. Chen, "How many annotations do we need for generalizing new-coming shadow images?" *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 11, pp. 6213–6224, Nov. 2023.

[6] A. Ecins, C. Fermüller, and Y. Aloimonos, "Shadow free segmentation in still images using local density measure," in *Proc. IEEE Int. Conf. Comput. Photogr. (ICCP)*, May 2014, pp. 1–8.

[7] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati, "Detecting moving objects, ghosts, and shadows in video streams," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 10, pp. 1337–1342, Oct. 2003.

[8] Z. Chen et al., "Triple-cooperative video shadow detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 2715–2724.

[9] S. Nadimi and B. Bhanu, "Physical models for moving shadow and object detection in video," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 8, pp. 1079–1087, Aug. 2004.

[10] G. D. Finlayson, S. D. Hordley, C. Lu, and M. S. Drew, "On the removal of shadows from images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 1, pp. 59–68, Jan. 2006.

[11] X. Huang, G. Hua, J. Tumblin, and L. Williams, "What characterizes a shadow boundary under the sun and sky?" in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 898–905.

[12] J.-F. Lalonde, A. A. Efros, and S. G. Narasimhan, "Detecting ground shadows in outdoor consumer photographs," in *Proc. Eur. Conf. Comput. Vis.* Heraklion, Greece: Springer, 2010, pp. 322–335.

[13] J. Zhu, K. G. G. Samuel, S. Z. Masood, and M. F. Tappen, "Learning to recognize shadows in monochromatic natural images," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 223–230.

[14] S. H. Khan, M. Bennamoun, F. Sohel, and R. Togneri, "Automatic feature learning for robust shadow detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1939–1946.

[15] T. F. Y. Vicente, L. Hou, C.-P. Yu, M. Hoai, and D. Samaras, "Large-scale training of shadow detectors with noisily-annotated shadow examples," in *Proc. Eur. Conf. Comput. Vis.* Amsterdam, The Netherlands: Springer, 2016, pp. 816–832.

[16] V. Nguyen, T. F. Y. Vicente, M. Zhao, M. Hoai, and D. Samaras, "Shadow detection with conditional generative adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4510–4518.

[17] X. Hu, L. Zhu, C.-W. Fu, J. Qin, and P.-A. Heng, "Direction-aware spatial context features for shadow detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7454–7462.

[18] L. Zhu et al., "Bidirectional feature pyramid network with recurrent attention residual modules for shadow detection," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 121–136.

[19] X. Hu, C.-W. Fu, L. Zhu, J. Qin, and P.-A. Heng, "Direction-aware spatial context features for shadow detection and removal," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 11, pp. 2795–2808, Nov. 2020.

[20] Q. Zheng, X. Qiao, Y. Cao, and R. W. H. Lau, "Distraction-aware shadow detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5162–5171.

[21] S. Hu, H. Le, and D. Samaras, "Temporal feature warping for video shadow detection," 2021, *arXiv:2107.14287*.

[22] X. Ding, J. Yang, X. Hu, and X. Li, "Learning shadow correspondence for video shadow detection," in *Computer Vision—ECCV*, S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, Eds., Cham, Switzerland: Springer, 2022, pp. 705–722.

[23] X. Lu et al., "Video shadow detection via spatio-temporal interpolation consistency training," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 3106–3115.

[24] Z. Chen, X. Lu, L. Zhang, and C. Xiao, "Semi-supervised video shadow detection via image-assisted pseudo-label generation," in *Proc. 30th ACM Int. Conf. Multimedia*. New York, NY, USA: Association for Computing Machinery, Oct. 2022, pp. 2700–2708, doi: 10.1145/3503161.3548074.

[25] Y. Wang, W. Zhou, Y. Mao, and H. Li, "Detect any shadow: Segment anything for video shadow detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 34, no. 5, pp. 3782–3794, May 2024.

[26] G. D. Finlayson, M. S. Drew, and C. Lu, "Entropy minimization for shadow removal," *Int. J. Comput. Vis.*, vol. 85, no. 1, pp. 35–57, Oct. 2009.

[27] Z. Chen, L. Zhu, L. Wan, S. Wang, W. Feng, and P.-A. Heng, "A multi-task mean teacher for semi-supervised shadow detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 5611–5620.

[28] X. Hu, T. Wang, C.-W. Fu, Y. Jiang, Q. Wang, and P.-A. Heng, "Revisiting shadow detection: A new benchmark dataset for complex world," *IEEE Trans. Image Process.*, vol. 30, pp. 1925–1934, 2021.

[29] L. Zhu, K. Xu, Z. Ke, and R. W. H. Lau, "Mitigating intensity bias in shadow detection via feature decomposition and reweighting," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 4702–4711.

[30] J. C. S. Jacques, C. R. Jung, and S. R. Musse, "Background subtraction and shadow detection in grayscale video sequences," in *Proc. 18th Brazilian Symp. Comput. Graph. Image Process. (SIBGRAPI)*, 2005, pp. 189–196.

[31] C. Benedek and T. Sziranyi, "Bayesian foreground and shadow detection in uncertain frame rate surveillance videos," *IEEE Trans. Image Process.*, vol. 17, no. 4, pp. 608–621, Apr. 2008.

[32] L. Liu et al., "SCOTCH and SODA: A transformer video shadow detection framework," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 10449–10458.

[33] X. Lu, W. Wang, C. Ma, J. Shen, L. Shao, and F. Porikli, "See more, know more: Unsupervised video object segmentation with co-attention Siamese networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3618–3627.

[34] L. Xi, W. Chen, X. Wu, Z. Liu, and Z. Li, "Online unsupervised video object segmentation via contrastive motion clustering," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 34, no. 2, pp. 995–1006, Feb. 2024.

[35] M. Sun, J. Xiao, E. G. Lim, C. Zhao, and Y. Zhao, "Unified multi-modality video object segmentation using reinforcement learning," *IEEE Trans. Circuits Syst. Video Technol.*, early access, Jun. 8, 2023, doi: 10.1109/TCSVT.2023.3284165.

[36] Y. Lu et al., "Label-efficient video object segmentation with motion clues," *IEEE Trans. Circuits Syst. Video Technol.*, early access, Jul. 25, 2024, doi: 10.1109/TCSVT.2023.3298853.

[37] Q. Qi, T. Hou, Y. Yan, Y. Lu, and H. Wang, "TCNet: A novel triple-cooperative network for video object detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 8, pp. 3649–3662, Aug. 2023.

[38] S. W. Oh, J.-Y. Lee, N. Xu, and S. J. Kim, "Video object segmentation using space-time memory networks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9226–9235.

[39] X. Lu, W. Wang, M. Danelljan, T. Zhou, J. Shen, and L. Van Gool, "Video object segmentation with episodic graph memory networks," in *Proc. Europeon Conf. Comput. Vis.* Glasgow, U.K.: Springer, 2020, pp. 661–679.

[40] H. Seong, J. Hyun, and E. Kim, "Kernelized memory network for video object segmentation," in *Proc. Eur. Conf. Comput. Vis.* Glasgow, U.K.: Springer, 2020, pp. 629–645.

[41] B. Duke, A. Ahmed, C. Wolf, P. Aarabi, and G. W. Taylor, "SSTVOS: Sparse spatiotemporal transformers for video object segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 5908–5917.

[42] Z. Yang, Y. Wei, and Y. Yang, "Associating objects with transformers for video object segmentation," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 34, 2021, pp. 1–12.

[43] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.

[44] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI Blog*, vol. 1, no. 8, p. 9, 2019.

[45] G. Synnaeve et al., "End-to-end ASR: From supervised to semi-supervised learning with modern architectures," 2019, *arXiv:1911.08460*.

[46] A. Dosovitskiy et al., "An image is worth 16×16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.

[47] A. Vaswani, P. Ramachandran, A. Srinivas, N. Parmar, B. Hechtman, and J. Shlens, "Scaling local self-attention for parameter efficient visual backbones," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 12894–12904.

[48] Z. Liu et al., "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 10012–10022.

[49] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. Eur. Conf. Comput. Vis.* Glasgow, U.K.: Springer, 2020, pp. 213–229.

[50] Y. Wang et al., "End-to-end video instance segmentation with transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 8741–8750.

[51] X. Hu et al., "Transformer tracking via frequency fusion," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 34, no. 2, pp. 1020–1031, Feb. 2024.

[52] J. Ye, C. Qiu, and Z. Zhang, "SNR-prior guided trajectory-aware transformer for low-light video enhancement," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 34, no. 3, pp. 1873–1885, Mar. 2024.

[53] N. Parmar et al., "Image transformer," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 4055–4064.

[54] Z. Teed and J. Deng, "RAFT: Recurrent all-pairs field transforms for optical flow," in *Proc. Eur. Conf. Comput. Vis.* Glasgow, U.K.: Springer, Aug. 2020, pp. 402–419.

[55] T. Y. Lin, P. Dollàr, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 2117–2125.

[56] C. Doersch, A. Gupta, and A. A. Efros, "Unsupervised visual representation learning by context prediction," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1422–1430.

[57] N. Komodakis and S. Gidaris, "Unsupervised representation learning by predicting image rotations," in *Proc. Int. Conf. Learn. Represent.*, 2018.

[58] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 30, 2017, pp. 1–11.

[59] H. Fan et al., "Multiscale vision transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 6824–6835.

[60] 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.

[61] Q. Hou, M.-M. Cheng, X. Hu, A. Borji, Z. Tu, and P. Torr, "Deeply supervised salient object detection with short connections," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3203–3212.

[62] Z. Deng et al., "R^3Net: Recurrent residual refinement network for saliency detection," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Palo Alto, CA, USA, Jul. 2018, pp. 684–690.

[63] H. Song, W. Wang, S. Zhao, and J. Shen, "Pyramid dilated deeper ConvLSTM for video salient object detection," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 715–731.

[64] H. Li, G. Chen, G. Li, and Y. Yu, "Motion guided attention for video salient object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 7273–7282.

[65] P. Voigtlaender, Y. Chai, F. Schroff, H. Adam, B. Leibe, and L.-C. Chen, "FEELVOS: Fast end-to-end embedding learning for video object segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 9481–9490.

[66] X. Fang, X. He, L. Wang, and J. Shen, "Robust shadow detection by exploring effective shadow contexts," in *Proc. 29th ACM Int. Conf. Multimedia*. New York, NY, USA: Association for Computing Machinery, Oct. 2021, pp. 2927–2935, doi: 10.1145/3474085.3475199.

[67] C. Huynh, A. T. Tran, K. Luu, and M. Hoai, "Progressive semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 16750–16759.

[68] D. Nilsson and C. Sminchisescu, "Semantic video segmentation by gated recurrent flow propagation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6819–6828.

[69] V. Verma et al., "Interpolation consistency training for semi-supervised learning," *Neural Netw.*, vol. 145, pp. 90–106, Jan. 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0893608021003993

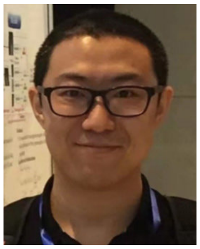
[70] A. Howard, A. Zhmoginov, L.-C. Chen, M. Sandler, and M. Zhu, "Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation," in *Proc. CVPR*, 2018, pp. 4510–4520.

[71] W.-S. Lai, J.-B. Huang, O. Wang, E. Shechtman, E. Yumer, and M.-H. Yang, "Learning blind video temporal consistency," in *Computer Vision—ECCV*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds., Cham, Switzerland: Springer, 2018, pp. 179–195.

Junhao Lin received the master's degree in computer science from Xiamen University, Xiamen, China, in 2023. He is currently a Research Assistant with The Hong Kong University of Science and Technology (Guangzhou). His research interests include video object detection and segmentation.

Jiaxing Shen received the B.E. degree in software engineering from Jilin University, Changchun, China, in 2014, and the Ph.D. degree in computer science from The Hong Kong Polytechnic University in 2019. He was a Visiting Scholar with the Media Laboratory, Massachusetts Institute of Technology, in 2017. He is currently an Assistant Professor with the School of Data Science, Lingnan University. His research has been published in top-tier journals, such as IEEE Transactions on Mobile Computing, ACM TOIS, ACM IMWUT, and IEEE Transactions on Knowledge and Data Engineering. His research interests include mobile computing, data mining, and the IoT systems. He was awarded conference best paper twice, including one from IEEE INFOCOM 2020.

Xin Yang (Member, IEEE) received the B.S. degree in computer science from Jilin University, Changchun, China, in 2007, and the joint Ph.D. degree in graphics from Zhejiang University, Hangzhou, China, and UC Davis, in July 2012. He is currently a Professor with the Department of Computer Science, Dalian University of Technology, Dalian, China. His research interests include computer graphics and robotic vision.



Huazhu Fu (Senior Member, IEEE) received the Ph.D. degree from Tianjin University in 2013. He is currently a Senior Scientist with the Institute of High Performance Computing (IHPC), A*STAR, Singapore. Prior to his current position, he was a Research Fellow with NTU, Singapore, from 2013 to 2015, a Research Scientist with I2R, A*STAR, from 2015 to 2018, and a Senior Scientist with the Inception Institute of Artificial Intelligence, United Arab Emirates, from 2018 to 2021. His research interests include computer vision, AI in healthcare, and trustworthy AI. He is a member of BISP TC. He received the Best Paper Award from ICME 2021. He has served as an Associate Editor for IEEE Transactions on Medical Imaging, IEEE Transactions on Neural Networks and Learning Systems, and IEEE Journal of Biomedical and Health Informatics, and the AC/Senior-PC for MICCAI, IJCAI, and AAAI.

Qing Zhang received the Ph.D. degree from the School of Computer Science, Wuhan University, Wuhan, China, in 2017. He is currently a Research Associate Professor with the School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China. His research interests include computer graphics, computer vision, and computational photography.

Ping Li (Member, IEEE) received the Ph.D. degree in computer science and engineering from The Chinese University of Hong Kong, Hong Kong, in 2013. He is currently an Assistant Professor with the Department of Computing and an Assistant Professor with the School of Design, The Hong Kong Polytechnic University, Hong Kong. He has published more than 200 top-tier scholarly research articles, pioneered several new research directions, and made a series of landmark contributions in his areas. He has an excellent research project reported by the ACM TechNews, which only reports the top breakthrough news in computer science worldwide. More importantly, however, many of his research outcomes have strong impacts to research fields, addressing societal needs and contributed tremendously to the people concerned. His current research interests include image/video stylization, colorization, artistic rendering and synthesis, computational art, and creative media.

Bin Sheng (Member, IEEE) received the Ph.D. degree in computer science and engineering from The Chinese University of Hong Kong, Hong Kong, in 2011. He is currently a Full Professor with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China. His current research interests include virtual reality and computer graphics. He is an Associate Editor of IEEE Transactions on Circuits and Systems for Video Technology.

Liansheng Wang (Member, IEEE) received the Ph.D. degree in computer science from The Chinese University of Hong Kong in 2012. He is currently an Associate Professor with the Department of Computer Science, Xiamen University, Xiamen, China. His research interests include medical image processing and analysis.

Lei Zhu (Member, IEEE) received the Ph.D. degree from the Department of Computer Science and Engineering, The Chinese University of Hong Kong. He is currently an Assistant Professor with the ROAS Thrust, The Hong Kong University of Science and Technology (Guangzhou), and an Assistant Professor in electronic and computer engineering with The Hong Kong University of Science and Technology. Before that, he was a Post-Doctoral Researcher with the Department of Applied Mathematics and Theoretical Physics (DAMTP), University of Cambridge. His research interests include computer graphics, computer vision, medical image processing, and deep learning.