Multi-Temporal Ultra Dense Memory Network For Video Super-Resolution

Peng Yi, Zhongyuan Wang, Kui Jiang, Zhenfeng Shao, and Jiayi Ma

***Abstract*—Video super-resolution (SR) aims to reconstruct the corresponding high-resolution (HR) frames from consecutive low-resolution (LR) frames. It is crucial for video SR to harness both inter-frame temporal correlations and intra-frame spatial correlations among frames. Previous video SR methods based on convolutional neural network (CNN) mostly adopt a single-channel structure and a single memory module, so they are unable to fully exploit inter-frame temporal correlations specific for video. To this end, this paper proposes a multi-temporal ultra dense memory (MTUDM) network for video super-resolution. Particularly, we embed convolutional long-short-term memory (ConvLSTM) into ultra dense residual block (UDRB) to construct an ultra dense memory block (UDMB) for extracting and retaining spatio-temporal correlations. This design also reduces the layer depth by expanding the width, thus avoiding training difficulties like gradient exploding and vanishing under a large model. We further adopt multi-temporal information fusion (MTIF) strategy to merge the extracted temporal feature maps in consecutive frames, improving the accuracy without requiring much extra computational cost. Experimental results on extensive public datasets demonstrate that our method outperforms the state-of-the-art methods by a large margin.**

***Index Terms*—Convolutional Neural Network, Video Super-Resolution, Ultra Dense Memory Block, Multi-Temporal Fusion.**

I. Introduction

Super-resolution (SR) has been a hot issue in computer vision fields for the past decades. SR refers to the technique recovering a high-resolution (HR) version from the given low-resolution (LR) image or video input, which is widely applied to many areas like video coding [1], [2], video surveillance [3], satellite imagery [4], human face hallucination [5], [6], etc.

There are various SR approaches, ranging from simple interpolation-based methods to complicated learning-based methods [7]–[12]. Traditional interpolation-based SR methods [13]–[15] focus on computing the unknown pixel values of HR space from given LR input. This kind of method is extremely fast owning to its little cost of computation. However, due to the lack of ability to learn extra samples, they are prone to produce over-blurry and distorted HR results. In recent years, deep-learning-based methods have dominated SR due to their impressive results [16]. Dong *et al.* [7] pioneered a three layer fully connected CNN, termed SRCNN, to approximate the complex non-linear mapping between the LR image and the HR counterpart. Benefiting from the end-to-end training strategy and the powerful learning capacity of deep neural networks, this method notably outperforms the conventional shallow counterparts (e.g., sparse-coding-based methods).

Compared to image SR, video SR puts more emphasis on exploiting the inter-frame temporal correlations between multiple consecutive frames rather than merely intra-frame spatial correlations within one frame. Most past video SR methods [17]–[21] are mainly engaged in motion and blur kernel estimation based on image prior to reconstruct HR frames. Inspired by CNN based image SR methods, a variety of deep-learning-based video SR approaches have also been proposed [9], [10], [22]–[26]. Huang *et al.* [22] extended SRCNN for video SR along the temporal dimension to capture the long term temporal dependency. Also based on SRCNN, Kappeler *et al.* [9] explored three different architectures for video SR. Caballero *et al.* [23] designed a framework incorporating motion estimation (ME), motion compensation (MC) and video SR, which is end-to-end trainable. Tao *et al.* [10] proposed a sub-pixel motion compensation (SPMC) layer, which not only compensates a frame to the reference frame, but also projects the compensated frame from LR space to the HR space. However, it has to use a shallow and simple network due to the heavy complexity of SPMC layer, which thus in turn hinders the temporal feature screening and the fusion between multiple frames.

In order to improve performance while reducing the network redundancy and avoiding gradient exploding and vanishing caused by increased network depth, we propose a multi-temporal ultra dense memory (MTUDM) video super-resolution network. Our model includes optical flow network and image-reconstruction network, which are respectively responsible for estimating motion between frames and representing spatio-temporal features. In particular, we devise an ultra dense memory block (UDMB) to retain temporal correlations between consecutive frames by incorporating ConvLSTM [27] into ultra dense residual block (UDRB). Our proposed UDMB also intends to extract hierarchical information from multiple channels and alleviate computational burden for each channel

This work is supported by National Key R&D Project (2016YFE0202300), National Natural Science Foundation of China (61671332, U1736206, 61773295), Hubei Province Technological Innovation Major Project (2017AAA123) and Natural Science Fund of Hubei Province (2018CFA024). *(Corresponding author: Zhongyuan Wang.)*

P. Yi, Z. Wang and K. Jiang are with the National Engineering Research Center for Multimedia Software, School of Computer Science, Wuhan University, Wuhan, 430072, China (e-mail: yipeng@whu.edu.cn; wzy_hope@163.com; kuijiang@whu.edu.cn).

Z. Shao is with the State Key Laboratory for Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan, 430072, China (e-mail: shaozhenfeng@whu.edu.cn).

J. Ma is with the Electronic Information School, Wuhan University, Wuhan, 430072, China (e-mail: jyma2010@gmail.com).

by reducing the network depth. Since the temporal correlations between frames is used only once in the previous network [10], SR construction cannot fully benefits from spatio-temporal complementarity. Therefore, a multi-temporal information fusion strategy is further adopted for better utilization of temporal information passed through preceding steps, which proves to improve the performance of our model when taking more than 3 frames as input. Experimental results on public datasets confirm the effectiveness of our method against the state-of-the-art counterparts in Section V.

In summary, our major contributions are highlighted as follows:

1) We establish a multi-temporal ultra dense memory (M-TUDM) video super-resolution network with a shallow depth but a broad width.
2) We propose an ultra dense memory block (UDMB) to retain temporal correlations and extract hierarchical information from multiple channels as well.
3) We present a multi-temporal information fusion strategy to make full use of historical information.

The rest of this paper is arranged as follows. In Section II, we survey image and video SR methods based on CNN. In Section III, we elaborate the proposed multi-temporal ultra dense memory network for video SR. In Section IV, we present the functions and analysis of different modules adopted in the network and how we explore the best model. In Section V, we give numerical and subjective results on public datasets. Section VI draws a conclusion.

II. Related Work

In this section, we first introduce single image super-resolution (SISR) methods and then survey video SR. Here we only focus on deep CNN based methods, which have shown impressive performance for image or video SR. Finally, we particularly summarize typical CNN's structures as well as their influence on SR.

A. CNN based Single Image SR

As a common basis for SR, SISR methods can inspire video SR regardless of the temporal property. Since there is only one LR image as the input to reconstruct the corresponding HR image, the SISR methods can only benefit from spatial correlations within one single image. Recently, with the popularity of deep learning, a lot of learning-based, especially CNN based models for SISR have been proposed [7], [8], [28]–[32], which have achieved more realistic results compared to early shallow learning methods. This kind of method is devoted to learning the mapping relationship from LR space to HR space.

Dong *et al.* [7] first proposed a super-resolution convolutional neural network (SRCNN), which is consisted of three convolutional layers. Kim *et al.* [8] put forward a very deep convolutional network for super-resolution (VDSR), which adds the bicubically magnified LR image to the output of the network. In other words, instead of learning to produce HR image from given LR image directly, VDSR learns to generate the residual image between HR image and bicubically magnified LR image. This strategy makes the network easier to converge especially when the network is very deep. Shi *et al.* [30] raised a sub-pixel convolutional layer, which arranges the depth of the feature maps into its size. Compared to transposed convolution, this sub-pixel magnification strategy needs no extra parameters and thus runs faster. Lai *et al.* [28] proposed a Laplacian pyramid SR network (LapSRN) to gradually reconstruct the sub-band residuals of HR images at multiple pyramid levels. Zhang *et al.* [32] came up with a residual dense block (RDB) to extract abundant local features, which allows direct connections from the state of preceding RDB to all the layers of current RDB, leading to favorable performance for image SR.

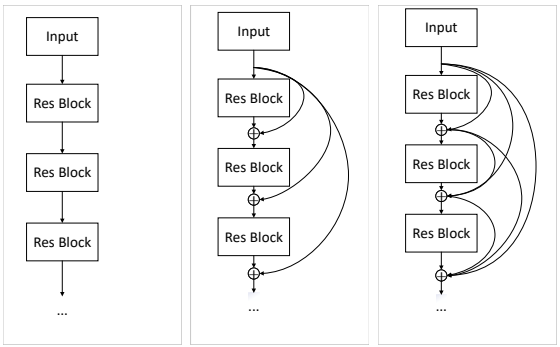
B. CNN based Video SR

Different from SISR, video SR adopts multiple consecutive video frames instead of a single image as input, thus the shared spatio-temporal redundancy can be used to constrain the mapping from LR space to HR space. Huang *et al.* [22] proposed a bidirectional recurrent convolutional network (BRCN), using bidirectional scheme, recurrent and conditional convolutions for temporal dependency modelling. Kappeler *et al.* [9] put forward a video super-resolution with convolutional neural networks (VSRnet) whcih is based on SRCNN, where they shared the weights in a symmetrical way to handle the input frames. Caballero *et al.* [23] advocated a video super-resolution with spatio-temporal networks and motion compensation (VESPCN), which extracts the optical flow in a coarse-to-fine manner, and uses spatial transformer for motion compensation. Tao *et al.* [10] proposed a detail-revealing deep video super-resolution (DRVSR), which utilizes a SPMC layer to extract sub-pixel information to achieve resolution enhancement. However, the SPMC layer requires large GPU memory cost but outperforms normal MC layer [23] little. Liu *et al.* [25] presented a robust video super-resolution with learned temporal dynamics (RVSR-LTD), which can adaptively determine the optimal scale of temporal dependency. In RVSR-LTD, the input frames are divided into multiple branches, and the last inference branch possesses complete input frames, while previous inference branches have only part of the input frames. In other words, different inference branches used in RVSR-LTD share a kind of redundancy, which may cause unnecessary waste of calculation. Sajjadi *et al.* [26] proposed a frame-recurrent video super-resolution (FRVSR), which takes last super-resolved frame to help reconstruct the current input frame. Compared to the methods [9], [10], [23], [25] using consecutive input frames to reconstruct the center frame, FRVSR has to process all input frames only once in testing. Unfortunately, FRVSR can never use the next frame to help reconstruct the current input frame.

Moreover, BRCN, VESPCN, VSRnet, DRVSR and RVSR-LTD only use a simple and direct way of connection, thus leading to shallow depth and limited performance. FRVSR only utilizes the previous frames while ignores the followed frames to super-resolve the current frame, however, the followed frames are of the same significance as the previous frames for video SR.

C. Connectivity Structure of CNN

In the research of neural networks, it has always been a significant issue on how to design a more effective structure. A variety of deep CNN variants have been developed in terms of network structures and optimization algorithms with the help of skip connection [33], [34]. Resnet [34] is able to effectively train networks with more than 100 layers by involving residual learning. DenseNet [35] concatenates shallow layers to the deep layers in the networks, trying to fully explore the advantages of skip connections. Further, these ideas have been tailored to support SISR and video SR, as mentioned as above.



(a) (b) (c)

Fig. 1. Distinct underlying connections in CNN. (a) Direct connection. (b) Skip connection. (c) Dense skip connection.

As shown in Figure 1(a), regular CNN based SR models simply use a serial of direct connected convolutional layers to extract the feature maps. They are naive and cannot reveal more realistic image details. In contrast, skipping connections between layers make neural networks deeper and perform better. Two representative skip connection schemes are shown in Figure 1(b) and 1(c). Especially, the dense connectivity pattern allows the network to reuse and bypass existing features from prior layers and ensures high accuracies in later layers. A two-dimensional multi-scale DenseNet has been further proposed by Huang *et al.* [36] to maintain coarse and fine level features all-throughout the network. The main idea is to enable multiple prediction exits in a network. For simple images, the results are obtained directly from the previous exit, while for complex images, the latter layers are used to obtain reliable results. However, this network only takes into account scale characteristic (coarse or fine) within a single image without considering the correlations between successive sequence images.

D. Motion Estimation and Compensation

Motion estimation and compensation in optical flow network is used to represent the temporal correlations in video SR. There appear some networks for motion estimation [37]–[39], but they are practically too complex. For example, FlowNetC [37] requires about 40 M parameters, and FlowNet2 [39] takes even more than 100 M parameters. Compensating the motion of input images with a total variation (TV)-based optical flow algorithm shows superiority. Jaderberg *et al.* [40] proposed a spatial transformer networks, where a differentiable layer warps images according to predicted affine transformation parameters. Caballero *et al.* [23] proposed a motion estimation and compensation scheme, based on spatial transformer networks, it can be trained along with the SR network. Accordingly, Tao *et al.* [10] proposed a SPMC layer and used ConvLSTM in a CNN framework, intended for fusing image information from aligned frames.

III. Our Method

In this section, we present the methodology for our multi-temporal ultra dense memory video super-resolution network, including the overall framework and details on major modules.

A. Framework of MTUDM Network

As shown in Figure 2, our model consists of two parts: optical flow network and image-reconstruction network. Video SR model aims to predict one HR frame from a serial of adjacent LR frames, and we thereby use the optical flow network to estimate the motion between frames. Generally, the motion estimation module inputs two frames to generate an optical flow vector field as follows:

$$F_{i\to j} = (u_{i\to j}, v_{i\to j}) = \mathrm{ME}(I_i, I_j; \theta_{ME}), \tag{1}$$

where $F_{i\to j}$ denotes the optical flow field generated from input frame I_i to I_j, $\mathrm{ME}(\cdot)$ is the motion estimation module, and θ_{ME} is the parameter for $\mathrm{ME}(\cdot)$. The motion estimation module is learnt from [23] owing to its involved fewer parameters, namely, 53 K.

Then, we use the optical flow for motion compensation and transforming the input LR frames into warped frames. Motion compensation intends to compensate the current frame to the reference frame, which can be described as follows:

$$J = \mathrm{MC}(I, F; \theta_{MC}), \tag{2}$$

where J denotes the compensated frame, $\mathrm{MC}(\cdot)$ represents the module for motion compensation, I is the input frame, F means the optical flow field, and θ_{MC} is the parameter for $\mathrm{MC}(\cdot)$. We have learned two approaches for motion compensation (MC): normal MC from [23] and SPMC from [10]. A normal MC only compensates the LR frame to the reference frame, while a SPMC compensates and magnifies the LR frame to a warped HR frame simultaneously. We choose normal MC, which performs only a little worse than SPMC but requires much less GPU memory.

After that, we send these warped LR frames into the image-reconstruction network. In the image-reconstruction network, successive residual blocks (UDMBs and UDRBs) are first used to extract feature information among frames, in which feature maps from different channels are fused to enhance the residual profile. The structure of these residual blocks is further illustrated in Figure 3, which is composed of UDMBs and a series of symmetrically assembled UDRBs. Then, the MTIF module is adopted to reserve the feature maps of all input frames, in which feature maps from preceding frames (temporally) are fused, further enhancing the memory ability of the network. The sub-pixel magnified residual map is finally added to the bicubically magnified LR input to generate

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2019.2925844, IEEE Transactions on Circuits and Systems for Video Technology

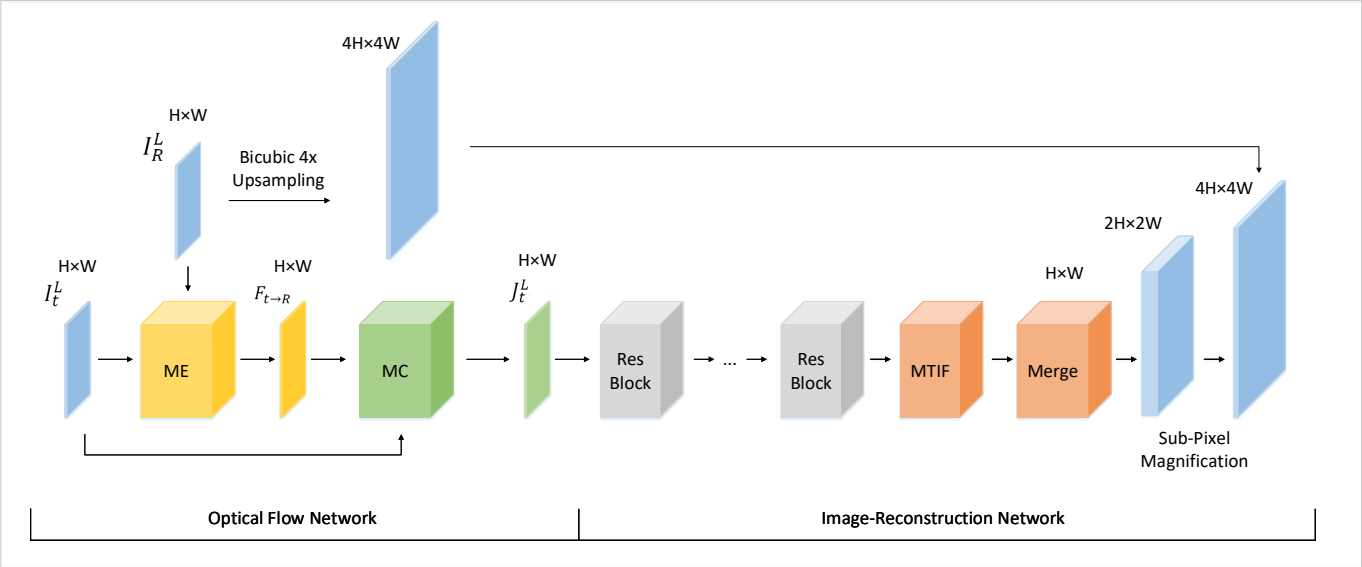


Fig. 2. The structure of our MTUDM SR network when upscaling factor is 4. I_t^L denotes LR frames, I_R^L represents the reference frame, $F_{t\to R}$ is the optical flow field, and J_t^L is LR frame warped by motion compensation. Our network manages to learn the residual image and then add it to the bicubically magnified LR input to obtain the final SR result.

the SR output. We elaborate these modules in the following respectively.

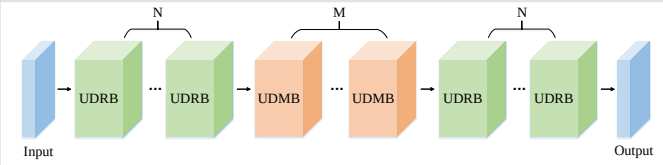


Fig. 3. Structure about the "Res Blocks" part of the image-reconstruction network in Figure 2. N and M denote the number of different residual blocks adopted in the network, and we keep same number of UDRBs before and after the UDMBs to form a symmetric structure.

B. Ultra Dense Memory Block

As shown in Figure 4, we extend the idea of DenseNet in [35], [36] and design an UDMB, which is composed of a sophisticated structure for better preservation of temporal motion information. Our idea of designing this block is to solve the problem caused by the increased depth in CNN, that is, the simple stacking of convolutional layers may lead to explosion or disappearing in the gradient, difficulty in convergence, degradation in performance, etc. UDMB shares similar structure with UDRB, except that in UDRB, the ConvLSTM layers are replaced of normal convolutional layers.

As illustrated in Figure 4, UDMB first conducts C different channels of convolutions on the input, which is supposed to extract hierarchical feature maps from one single input, and get C paralleled path of feature information. Then, feature information from different channels is convoluted respectively and extracted rather self-indenpendent feature maps, which is intended to utilize the self-identity of each channel. Moreover, feature maps from different channels are merged together, along with the preceding feature maps from each channel. This concatenation is used to utilize information from paralleled channels and self's past. Note that we use convolution with 1×1 kernel to handle the merged feature maps and reduce the numbers of feature maps to 64, for the purpose of releasing parameters and computational complexity, as well as extracting more concise information. This operation can be considered a weight adjustment and fusion of information from different channels.

3D-convolution [41] may provide longer-term temporal correlations capability by proactively expanding the time-domain dimension of 3D-convolution kernels. Nevertheless, it requires huge computational cost due to additional convolution across the temporal dimension. More importantly, our proposed ultra dense memory block (UDMB) is based on memory mechanism, but 3D CNN does not explicitly provide such mechanism. Therefore, we introduce ConvLSTMs to build up our UDMB, which treats frames as fluids that flow sequentially through the network. As one LR frame goes through the UDMB, the cell of ConvLSTM layer is supposed to keep feature information of this frame. The reserved information from last frame is processed through functions in ConvLSTM and merged with current input frame information. This way, ConvLSTM learns to forget the temporal-irrelevant information and remember the temporal-related information, which is suitable for processing continuous video frames.

C. Multi-Temporal Information Fusion

In previous work of Tao *et al.* [10], because the temporal correlations between frames is exploited only once throughout the network, the network cannot fully benefit from spatio-temporal correlated information. Therefore, we propose a MTIF module for better utilization of temporal information passed through previous steps. As discussed above, UDRB intends to exploit intra-frame spatial correlations, while UDMB is specifically used for capturing inter-frame correlations.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2019.2925844, IEEE Transactions on Circuits and Systems for Video Technology

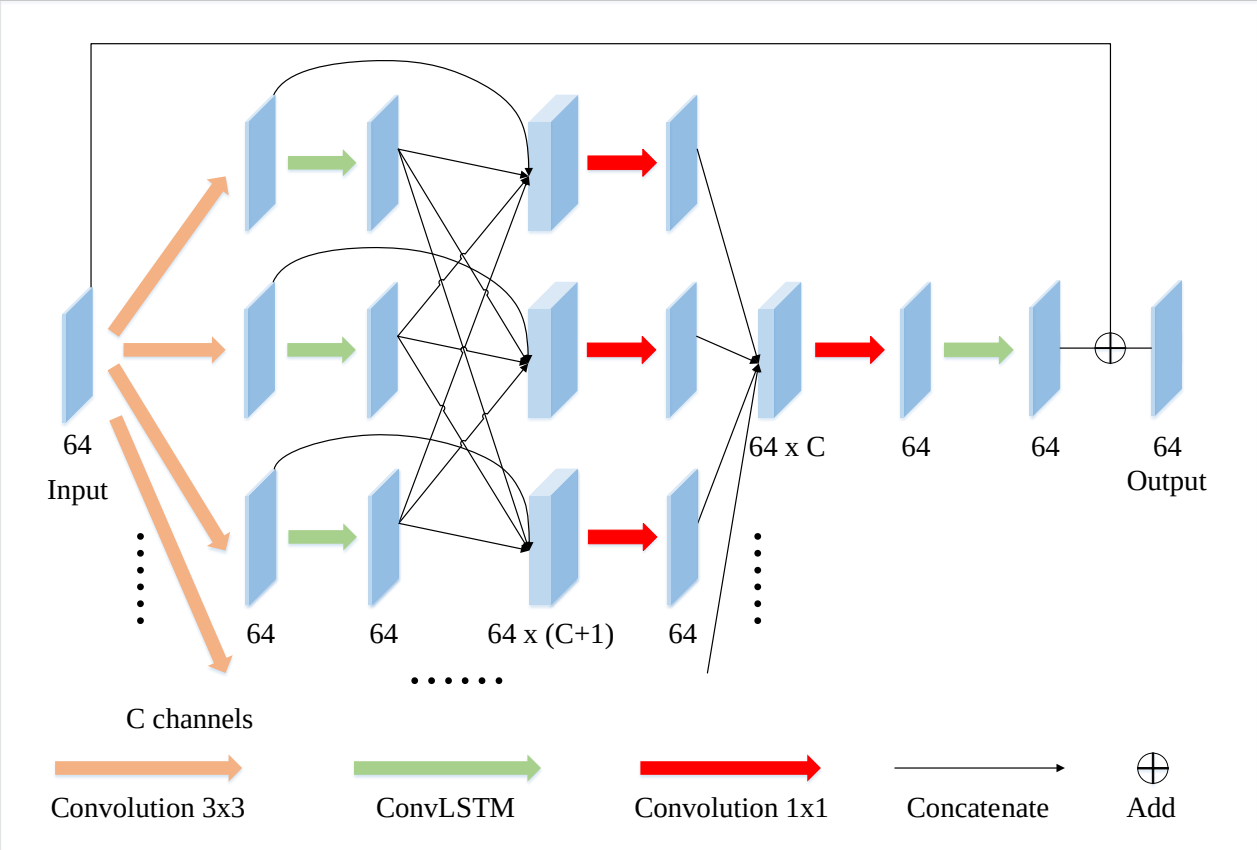


Fig. 4. Structure of the proposed ultra dense memory block (UDMB). Input feature maps are first conducted by a certain number (denoted as C and $C \geq 1$) of different channels of convolutional layers. Feature maps from different channels are then processed by ConvLSTM layers, whose information will be preserved in the cells of ConvLSTM. Later, feature maps from channels and the previous step are concatenated together and merged by 1×1 convolution. Further, the feature maps concatenated from different channels are merged by a 1×1 convolutional layer. Last, the merged feature maps are passed through a ConvLSTM layer and added to the original input feature maps, resulting in the output. Note that the depth of these feature maps are listed under them, e.g. 64, $64 \times C$ and $64 \times (C + 1)$.

Therefore, it is reasonable to combine them for full use of spatio-temporal correlations. Normally, after passing through a serial of UDRBs and UDMBs, the feature information of a frame has been exploited and will be sent to the merge module for sub-pixel magnification. Without MTIF, it only relies on the ConvLSTM to reserve the multi-frame temporal correlations. By adopting MTIF strategy, as shown in Figure 2, where MTIF module reserves the feature map of a frame at one time. As frames pass through the network, the MTIF module will receive and reserve all feature information from previous frames. The retained multi-temporal feature information is further merged and magnified to reconstruct the HR frame. In this way, the network not only benefits much from ConvLSTM layer, but also leverages MTIF to capture inter-frame temporal correlations. Formally, this process can be described as follows:

$$H_{i+1} = \{H_i, I_i\}, \tag{3}$$

where H_i denotes the feature maps of the i^{th} step reserved in the MTIF module, and I_i represents the input feature maps of the i^{th} step to the MTIF module. This strategy shares similar idea with dense skip connections proposed by Huang *et al.* [35], whereas, feature maps of time series data (video frames) are handled and saved. Information from different frames can be fully connected and fused, so-called multi-temporal information fusion. Subsequent experimental results will show the superiority of our multi-temporal structure.

IV. Ablation Studies

In this section, we thoroughly investigate our established model so as to figure out the most optimal model configuration, including implementation details, validation and analysis of individual modules adopted in the model.

A. Data Collection

For SISR, there are a lot of public datasets like DIV2K [42]. For video SR, the datasets are required to be high-definition and with little noise, but there are few datasets satisfied. Thus, we decided to collect video frames from HD videos, mostly from HD documentaries, since documentaries are less post processed and more realistic than commercial movies. We collected 542 sequences (each is composed of 32 consecutive frames) from HD documentaries with various kinds of scenes like forest, snow, desert, urban life, etc. We randomly choose 522 sequences for training and 20 for validation during the training. Following a lot of previous methods [9], [10], [23]–[25], the original video frames are down-sampled bicubically

to generate the LR frames. We set the size of input LR frames as 32×32, and batch size is set as 10.

B. Implementation Details

We adopt L1 loss function for optical flow network and Charbonnier loss function [28], [43] for image-reconstruction network in training. Mathematically, the loss functions are described as follows:

$$\mathcal{L}_{IR} = \sum_{i=-T}^{T} \lambda_i \sqrt{(I_0^H - \mathrm{SR}(J_i))^2 + \epsilon^2}. \tag{4}$$

$$\mathcal{L}_{OF} = \sum_{i=-T}^{T} \left\| I_i^L - \tilde{I}_{0\to i}^L \right\|_1 + \alpha \left\| \bigtriangledown F_{i\to 0} \right\|_1. \tag{5}$$

$$\mathcal{L} = \mathcal{L}_{IR} + \beta \mathcal{L}_{OF}. \tag{6}$$

$\mathcal{L}_{IR}$ and $\mathcal{L}_{OF}$ represent loss of image-reconstruction and optical flow network respectively. In Equation (4), I_0^H represents the center HR frame, while J_i denotes LR frame of the i^{th} time step after motion compensation as described in Equation (2). λ_i is the weight of the i^{th} time step. In our network, later frames can receive the information from previous frames by introducing ConvLSTMs, and it is more important to reconstruct further frames. Thus, inspired from DRVSR [10], we set $\lambda_{-T} = 0.5$, while $\lambda_T = 1.0$ and we linearly interpolate its intermediate values to make λ_i an incremental arithmetic progression. The λ_i is further normalized by $\lambda_i = \frac{\lambda_i}{\sum_{i=-T}^{T} \lambda_i}$. $\mathrm{SR}(\cdot)$ represents the function of super-resolution, and ϵ is a small constant (empirically set to 10^{-3}) to make this loss function differentiable. Actually this Charbonnier loss is a differentiable variant of L1 norm.

In Equation (5), I_i^L is the i^{th} LR frame, and $\tilde{I}_{0\to i}^L$ represents backward warped I_0^L according to $F_{i\to 0}$, while $\bigtriangledown F_{i\to 0}$ denotes the total variation on (u, v) of $F_{i\to 0}$ as described in Equation (1). Because $\left\| \bigtriangledown F_{i\to 0} \right\|_1$ is a penalty term, we use α to adjust its weight in the equation. Similarly, we adopt β to adjust the weight of $\mathcal{L}_{OF}$, and we set $\alpha = \beta = 0.01$, the same as [10].

The initial learning rate is set as 10^{-3}, which follows a Polynomial decay to 10^{-5} after 10^5 iterations, and Parametric Rectified Linear Unit (PReLU) [44] is adopted as activation for every convolutional layer, whose parameter is initialized as 0.2. During exploration, we train our model at $4\times$ upscaling, then train our best model at $2\times$ and $3\times$ upscaling. The video frames are converted from RGB color space to YCbCr color space, and only the luminance channel is sent to the network.

We first train optical flow network for 10^4 iterations using $\mathcal{L}_{OF}$, then train image-reconstruction network for another 10^4 iterations using $\mathcal{L}_{IR}$. At last, these two networks are trained together with $\mathcal{L}$. We conduct experiments with an Intel I7-8700K CPU and an NVIDIA GTX 1080Ti GPU.

C. Model Simplification

We simplify DRVSR [10] and then improve it by introducing the proposed UDMB, UDRB and MTIF. The SPMC layer in DRVSR costs much GPU memory and has only a limited effect. We replace the sub-pixel motion compensation (SPMC) layer with a normal MC layer [23] without pre-interpolation. Then, we adopt sub-pixel magnification [30] in place of transposed convolution, and reduce the depth of ConvLSTM layer from 128 to 64 to lower time and memory costs. The training process is shown in Figure 5(a). As shown, DRVSR-MC shows a slight drop on peak-signal-to-noise-ratio (PSNR) compared to DRVSR, while it has the same parameters but costs less time for training. Further, SBM (simplified base model) drops about another 0.21 dB on PSNR, while it has fewer parameters, and costs less time and GPU memory for training, which enables us to train a deeper network. Based on SBM, we will make improvements and accomplish promising results.

D. Validity of UDRB and UDMB

We first set the channel number of UDRB and UDMB as 3, and add them to the SBM model. As illustrated in Figure 5(a), SBM-R2C3 shows a substantial gain, surpassing SBM about 0.23 dB. By replacing one ConvLSTM layer with one UDMB, SBM-R2M1C3 surpasses DRVSR [10] about 0.24 dB. Note that these models are light and have no more than about 2 M parameters, and we just show these cases to prove the proposed residual blocks are effective even under light models.

Because the proposed residual blocks follow a multi-channel design, we then explore the influence of channel number C under larger models. As demonstrated in Figure 5(b), we change the channel number C from 1 to 5, and adjust the number of UDRBs accordingly to keep these models with roughly the same number of parameters (except model SBM-R100C1, which has about 9 M parameters). It can be seen that model SBM-R60C1 achieves the best performance, as it becomes a very deep (more than 200 layers) single-channel model. However, by stacking more UDRBs naively, model SBM-R100C1 actually behaves worse than SBM-R60C1, whcih indicates the bottleneck of the single-channel structure. In fact, we have also tried a deeper model SBM-R120C1, however, it becomes untrainable due to the gradient exploding caused by its depth. Besides, because SBM-R60C1 is a single-channel model, it is unable to enjoy the advantages of parallel computing, and runs slower than other multi-channel models like SBM-R18C3. As shown in Table I, SBM-R60C1 needs 1.42 s while SBM-R18C3 takes 1.23 s for testing.

From the observations above, we choose channel number C as 3 for further exploring. As demonstrated in Figure 5(c), we change the number of UDMBs from 1 to 4, and adjust the number of UDRBs accordingly to keep these models with roughly the same number of parameters (except model UDM-R8M4C3, which has about 10 M parameters). It is observed that model UDM-R6M3C3 already achieves a slightly better performance than SBM-R60C1, by adding more residual blocks, UDM-R8M4C3 achieves a much better performance. This phenomenon further indicates the superiority of the multi-channel design of UDRB and UDMB, because the single-channel model SBM-R100C1 behaves worse than its shallower counterpart SBM-R60C1. As demonstrated in Table I, model SBM-R60C1 needs 1.42 s while UDM-R6M3C3 takes only

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2019.2925844, IEEE Transactions on Circuits and Systems for Video Technology

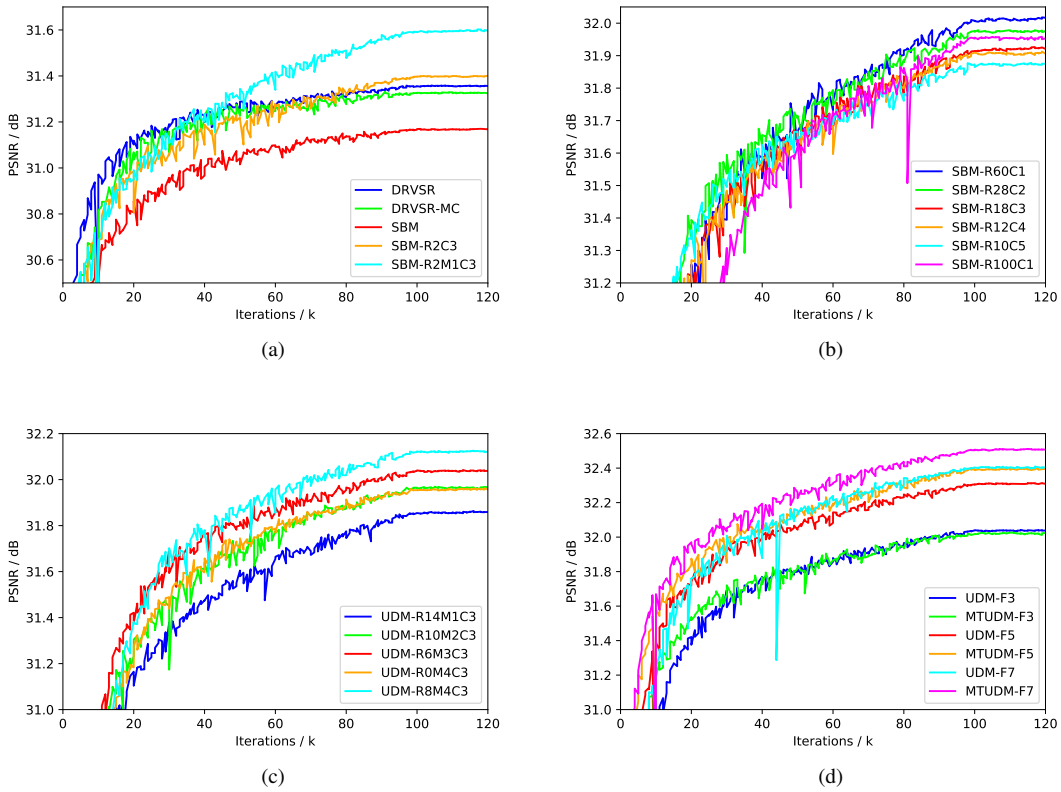


Fig. 5. Training processes for different models, where we omit training process of optical flow network. (a) DRVSR-MC denotes DRVSR using MC layer rather than SPMC layer, and SBM indicates the final simplified base model. SBM-R2C3 denotes SBM with two UDRBs around the ConvLSTM layer (one before and after), where "C3" indicates the channel number of UDRB is 3, and SBM-R2M1C3 represents SBM using one UDMB and two UDRBs around it. (b) SBM-R60C1 denotes SBM with 60 UDRBs around the ConvLSTM layer (30 before and 30 after), and the channel number C is set as 1. (c) UDM-R6M3C3 denotes a network consisting of 6 UDRBs and 3 UDMBs, and the channel number C is set as 3. (d) For simplicity, we denote UDM-R6M3C3 as UDM-F3 there, where "F3" means the input frame number as 3. MTUDM represents the model with MTIF strategy. The names of other models can be inferred in the same way.

TABLE I
COMPARISON OF DIFFERENT MODELS. FOR $4\times$ SR, PSNR IS EVALUATED ON 20 SEQUENCES COLLECTED FOR EVALUATION AND TESTING TIME IS MEASURED BY GENERATING ONE 1920×1080 VIDEO FRAME.

Metric	DRVSR	SBM-R2M1C3	SBM-R60C1	SBM-R100C1	SBM-R18C3	UDM-R6M3C3	UDM-F5	MTUDM-F5	UDM-F7
Parameter (M)	1.722	2.014	5.580	9.041	5.498	5.857	5.857	5.919	5.857
Testing time (s)	0.37	0.34	1.42	2.09	1.23	0.92	1.52	1.53	2.13
PSNR (dB)	31.36	31.60	32.02	31.96	31.93	32.04	32.31	32.40	32.41

0.92 s and achieves a slightly better performance, thus, we choose UDM-R6M3C3 for further exploring.

As discussed before, the input of UDRB and UDMB are processed by different channels of convolutional layers, which is supposed to extract independent and interdependent features among channels. We extract images from different channels in model UDM-R6M3C3 to show how the channels contribute to the SR task. As demonstrated in Figure 6, images on the top row are the outputs from three different channels. It can be observed that the second channel extracts most high-frequency image details, while the third channel is focused on capturing more low-frequency information, and the first channel makes a balance between them. Images from these three channels are merged together to form the final residual image, which contains most comprehensive information, from low-frequency information to high-frequency details. That is to say, three channels work together to obtain hierarchical information, and fuse complete levels of information. In conclusion, multiple channels of UDRB and UDMB alleviate the computational burden for each channel, as well as increase the width and reduce the depth of the network, which helps avoid the gradient exploding under a large model.

E. Validity of Multi-Temporal Information Fusion

At last, we explore the effectiveness of the MTIF module. As demonstrated in Figure 5(d), by adopting MTIF, MTUDM-F3 seems to behave worse than UDM-F3. This phenomenon accords to our expectation, because by taking only 3 frames

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2019.2925844, IEEE Transactions on Circuits and Systems for Video Technology

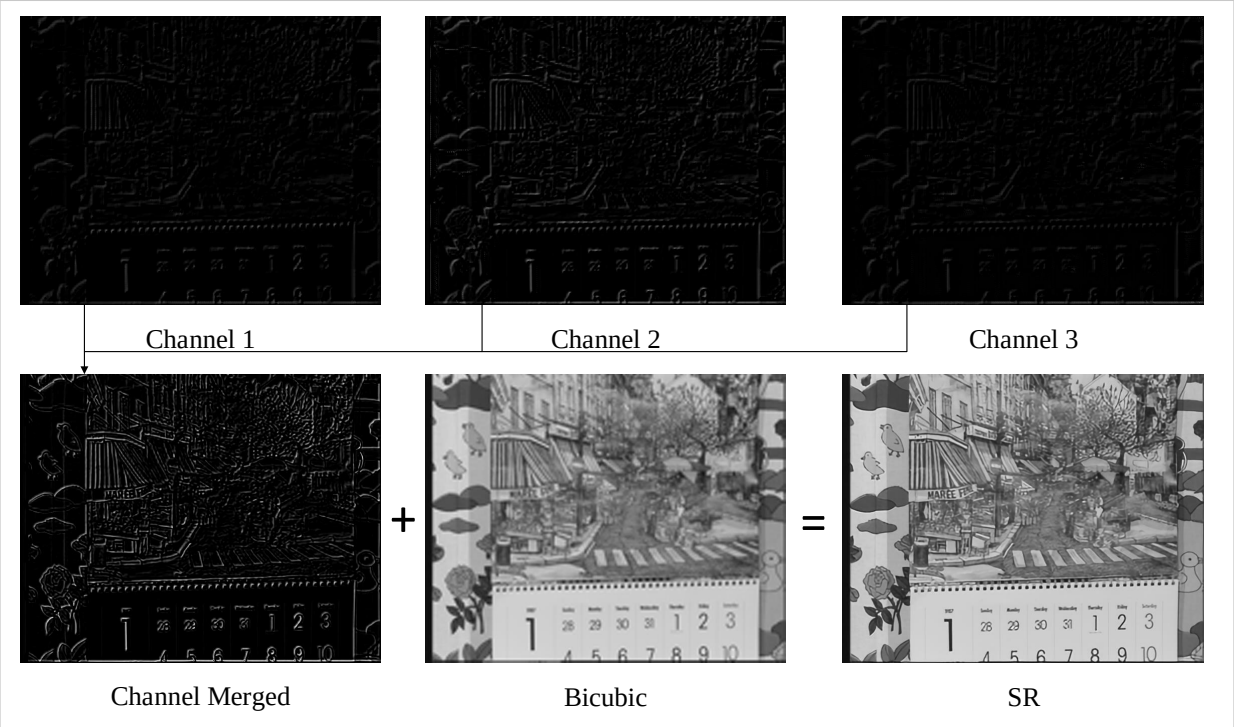


Fig. 6. Images from different channels in the network are extracted to show the function of UDRB and UDMB. Images on the top row are the outputs of three different channels in the network. Images on the bottom row are the output of the merged channels (also the residual image), bicubically magnified LR image and the final SR image, respectively. As shown in Figure 2, we add the residual image to the bicubically magnified LR input to obtain the SR result. Note that in order to obtain a more visible effect, pixel values of images from different channels and the residual image are multiplied by a factor of 2.

TABLE II
PSNR (DB)/SSIM OF DIFFERENT SR MODELS ON SINGLE IMAGE DATASETS. BEST PERFORMANCE IS SHOWN IN BOLD.

Methods	Scale	Set5	Set14	BSDS100	URBAN100	MANGA109
Bicubic		33.69 / 0.931	30.25 / 0.870	29.57 / 0.844	26.89 / 0.841	30.86 / 0.936
A+ [45]		36.60 / 0.955	32.32 / 0.906	31.24 / 0.887	29.25 / 0.895	35.37 / 0.968
SRCNN [7]		36.72 / 0.955	32.51 / 0.908	31.38 / 0.889	29.53 / 0.896	35.76 / 0.968
VDSR [8]	×2	**37.53 / 0.959**	33.05 / **0.913**	**31.90 / 0.896**	30.77 / **0.914**	37.22 / **0.975**
LapSRN [28]		37.52 / **0.959**	33.08 / **0.913**	31.80 / 0.895	30.41 / 0.910	37.27 / 0.974
DRVSR [10]		37.32 / 0.957	33.00 / 0.911	31.55 / 0.893	30.20 / 0.907	37.08 / 0.973
UDM-F3 (ours)		**37.53** / 0.958	**33.16 / 0.913**	31.71 / 0.895	**30.80 / 0.914**	**37.74** / 0.974
Bicubic		30.41 / 0.869	27.55 / 0.775	27.22 / 0.741	24.47 / 0.737	26.99 / 0.859
A+ [45]		32.62 / 0.909	29.15 / 0.820	28.31 / 0.785	26.05 / 0.799	29.93 / 0.912
SRCNN [7]		32.78 / 0.909	29.32 / 0.823	28.42 / 0.788	26.25 / 0.801	30.59 / 0.914
VDSR [8]	×3	33.67 / 0.921	29.78 / 0.832	**28.83 / 0.799**	**27.14** / 0.829	32.01 / 0.934
LapSRN [28]		33.82 / **0.922**	29.87 / 0.832	28.82 / 0.798	27.07 / 0.828	32.21 / **0.935**
DRVSR [10]		33.63 / 0.919	29.76 / 0.831	28.56 / 0.795	26.79 / 0.821	31.65 / 0.929
UDM-F3 (ours)		**33.94 / 0.922**	**29.97 / 0.834**	28.73 / **0.799**	27.13 / **0.830**	**32.30 / 0.935**
Bicubic		28.43 / 0.811	26.01 / 0.704	25.97 / 0.670	23.15 / 0.660	24.93 / 0.790
A+ [45]		30.32 / 0.860	27.34 / 0.751	26.83 / 0.711	24.34 / 0.721	27.03 / 0.851
SRCNN [7]		30.50 / 0.863	27.52 / 0.753	26.91 / 0.712	24.53 / 0.725	27.66 / 0.859
VDSR [8]	×4	31.35 / 0.883	28.02 / 0.768	27.29 / 0.726	25.18 / 0.754	28.83 / 0.887
LapSRN [28]		31.54 / 0.885	28.19 / 0.772	**27.32 / 0.727**	25.21 / 0.756	29.09 / 0.890
DRVSR [10]		31.40 / 0.881	28.05 / 0.768	27.04 / 0.722	25.00 / 0.750	28.48 / 0.881
UDM-F3 (ours)		**31.67 / 0.886**	**28.22 / 0.773**	27.15 / **0.727**	**25.31 / 0.761**	**29.10 / 0.891**

as input, the network is already able to capture the temporal correlations without the help of MTIF. However, when taking more frames as input, the network may fail to capture the long-range correlations by relying only on ConvLSTM layers. Thus, the MTIF module containing multi-temporal feature information is able to enhance the memory ability of the

TABLE III
PSNR (DB)/SSIM OF DIFFERENT SR MODELS FOR VIDEOSET4 TESTING DATASET. BEST PERFORMANCE IS SHOWN IN BOLD.

		SISR methods					
Scale	Metric	Bicubic	A+ [45]	SRCNN [7]	VDSR [8]	DRCN [46]	LapSRN [28]
x2	PSNR / SSIM	28.43 / 0.8676	30.53 / 0.9154	30.70 / 0.9172	31.44 / 0.9257	31.68 /0.9269	31.86 / 0.9290
x3	PSNR / SSIM	25.28 / 0.7329	26.36 / 0.7904	26.51 / 0.7933	26.82 / 0.8089	26.99 / 0.8122	26.95 / 0.8158
x4	PSNR / SSIM	23.79 / 0.6332	24.59 / 0.6889	24.69 / 0.6918	24.98 / 0.7119	25.03 / 0.7141	25.06 / 0.7170
		Video SR methods					
Scale	Metric	Bayesian [18]	VSRnet AMC [9]	VSRnet MC [9]	MCResNet [24]	DRVSR [10]	MTUDM-F5 (ours)
x2	PSNR / SSIM	29.69 / 0.9055	30.97 / 0.9217	31.30 / 0.9278	32.28 / 0.9433	32.50 / 0.9432	**33.80 / 0.9525**
x3	PSNR / SSIM	25.82 / 0.8323	26.61 / 0.8014	26.79 / 0.8098	27.54 / 0.8448	27.99 / 0.8606	**28.83 / 0.8860**
x4	PSNR / SSIM	25.06 / 0.7466	24.74 / 0.6986	24.84 / 0.7049	25.45 / 0.7467	25.90 / 0.7678	**26.57 / 0.7989**

TABLE IV
PSNR (DB) OF ONLY THE CENTER FRAME FROM EACH VIDEO SEQUENCE IN [47] ON DIFFERENT SISR AND VIDEO SR MODELS WHEN UPSCALING FACTOR IS 4. BEST PERFORMANCE IS SHOWN IN BOLD.

	VSRnet [9]	Deep-DE [47]	ESPCN [30]	VDSR [8]	LapSRN [28]	RVSR-LTD [25]	DRVSR [10]	MTUDM-F5 (ours)
calendar	20.99	21.40	20.97	21.50	21.73	21.61	22.53	**23.35**
city	24.78	25.72	25.60	25.16	25.18	**26.29**	25.98	26.23
foliage	23.87	24.92	24.24	24.41	24.40	24.99	25.30	**25.82**
penguin	35.93	30.69	36.50	36.60	36.78	36.68	36.92	**37.31**
temple	28.34	29.50	29.17	29.81	29.68	30.65	31.61	**33.41**
walk	27.02	26.67	27.74	27.97	28.10	28.06	28.54	**29.58**
average	26.82	26.48	27.29	27.58	27.65	28.05	28.48	**29.28**

network. As illustrated in Figure 5(d), MTUDM-F5 outperforms UDM-F5 about 0.09 dB. This increase seems to be trivial, however, note that MTUDM-F5 performs quite close to UDM-F7, while the former requires fewer input frames and runs faster. Besides, MTUDM-F5 requires only 1% more parameters (about 62 K) compared to UDM-F5, which is negligible. Significantly, MTUDM-F5 takes 1.53 s for testing, while UDM-F7 costs about 39% more time (2.13 s) to achieve the same performance. Therefore, it can be said that MTIF help the network to achieve the same performance with less time complexity. In all, these metrics have proven that MTIF is able to improve the performance while requiring little extra parameters and calculation cost.

V. EXPERIMENTAL RESULTS

We compare our model with both SISR models like SRCNN [7], VDSR [8], LapSRN [28], and video SR models like VSRnet [9], Deep-DE [47], ESPCN [30], MCResNet [24], RVSR-LTD [25] , DRVSR [10] and FRVSR [26]. We use UDM-F3 for SISR, because more input frames cannot contribute more to the SISR task. Model MTUDM-F5 is adopted for video SR, which is able to generate promising results with acceptable speed. We compute PSNR and SSIM as quantitative indicators, which are widely used to evaluate the model performance.

A. On Single Image Datasets

Although our model UDM-F3 is focused on super-resolving HR frames from multiple LR frames, it is sill able to generate a satisfactory result for single image. Thus, we decide to conduct comparison experiments on single image datasets. For SISR, we choose Set5 [49], Set14 [50], BSDS100 [51], Urban100 [52] and Manga109 [53] as test datasets. Note that UDM-F3 and DRVSR trained by us both take 3 adjacent frames to reconstruct the center frame. As there is only one image in single-image datasets, we send the counterparts of one image to simulate its adjacent frames. As illustrated in Table II, although our model is specifically designed for video SR and trained on video datasets, it is still able to achieve promising results on single images, benefiting from sophisticated structure of our model. Particularly, our model UDM-F3 is more robust in a larger upscaling factor, and achieves considerably better results than LapSRN [28], which is specifically designed for SISR and trained on SISR datasets, at the $4\times$ magnification.

B. On Video Datasets

For video SR, we first choose Videoset4 [54] as the testing dataset. Videoset4 testing dataset is composed of four scenes: *calendar*, *city*, *foliage* and *walk*. These four scenes respectively have 41, 34, 49 and 47 frames , whose resolution are in 720 $\times$ 576, 704$\times$576, 720$\times$480, and 720$\times$480, respectively. For evaluation, we skip the first and last two frames as [9], [24] do,

TABLE V
PSNR (DB) / SSIM OF DIFFERENT VIDEO SR MODELS ON VIDEO SEQUENCES FROM [48]. BEST PERFORMANCE IS SHOWN IN BOLD. NOTE THAT THESE MODELS ARE TRAINED ON OUR DATASETS.

Sequences	Scale	Bicubic	VESPCN [23]	RVSR-LTD [25]	MCResNet [24]	DRVSR [10]	FRVSR [26]	MTUDM-F5 (ours)
archpeople		38.65 / 0.9793	42.77 / 0.9887	42.79 / 0.9889	42.63 / 0.9883	42.75 / 0.9887	42.91 / 0.9889	**43.11 / 0.9894**
archwall		42.74 / 0.9777	45.69 / 0.9879	46.17 / 0.9885	45.89 / 0.9884	46.16 / 0.9893	46.25 / 0.9895	**46.91 / 0.9911**
auditorium		30.37 / 0.9328	34.80 / 0.9710	34.15 / 0.9678	34.92 / 0.9717	35.46 / 0.9742	35.64 / 0.9755	**36.49 / 0.9796**
band		36.45 / 0.9766	40.74 / 0.9884	40.60 / 0.9884	41.12 / 0.9890	41.43 / 0.9895	41.41 / 0.9891	**42.29 / 0.9909**
caffe		40.34 / 0.9822	45.86 / 0.9910	45.99 / 0.9912	46.33 / 0.9914	46.64 / 0.9919	47.30 / 0.9924	**47.91 / 0.9933**
camera	x2	49.04 / 0.9949	50.27 / 0.9952	51.56 / 0.9959	50.33 / 0.9952	50.02 / 0.9951	50.47 / 0.9954	**50.32 / 0.9953**
clap		37.26 / 0.9801	42.75 / 0.9903	42.56 / 0.9902	43.06 / 0.9905	43.52 / 0.9910	43.57 / 0.9910	**44.59 / 0.9926**
lake		33.70 / 0.9203	35.86 / 0.9565	35.80 / 0.9552	36.08 / 0.9581	36.70 / 0.9618	37.19 / 0.9656	**37.76 / 0.9709**
photography		38.83 / 0.9800	43.59 / 0.9906	43.59 / 0.9905	43.83 / 0.9910	44.20 / 0.9914	44.67 / 0.9922	**45.28 / 0.9930**
polyflow		36.59 / 0.9709	44.11 / 0.9891	43.87 / 0.9884	44.36 / 0.9898	44.86 / 0.9903	45.13 / 0.9903	**46.14 / 0.9925**
average		38.40 / 0.9695	42.64 / 0.9849	42.71 / 0.9845	42.85 / 0.9853	43.17 / 0.9863	43.45 / 0.9870	**44.08 / 0.9888**
archpeople		34.69 / 0.9515	38.22 / 0.9722	37.99 / 0.9711	38.20 / 0.9722	38.32 / 0.9725	38.48 / 0.9732	**39.26 / 0.9776**
archwall		38.55 / 0.9456	41.66 / 0.9705	41.94 / 0.9715	42.17 / 0.9735	42.84 / 0.9778	43.64 / 0.9806	**44.19 / 0.9834**
auditorium		27.37 / 0.8683	30.14 / 0.9253	29.82 / 0.9205	30.29 / 0.9269	30.84 / 0.9343	31.42 / 0.9414	**32.41 / 0.9514**
band		32.37 / 0.9419	36.13 / 0.9713	36.05 / 0.9703	36.43 / 0.9727	36.92 / 0.9750	37.18 / 0.9757	**38.01 / 0.9796**
caffe		35.35 / 0.9537	40.27 / 0.9776	40.79 / 0.9793	40.96 / 0.9798	42.14 / 0.9828	42.18 / 0.9834	**43.65 / 0.9860**
camera	x3	44.72 / 0.9908	46.53 / 0.9925	46.83 / 0.9929	46.63 / 0.9925	47.12 / 0.9927	47.75 / 0.9931	**48.65 / 0.9939**
clap		33.12 / 0.9490	37.87 / 0.9749	37.65 / 0.9735	38.13 / 0.9755	38.98 / 0.9786	39.35 / 0.9793	**40.36 / 0.9835**
lake		30.98 / 0.8397	32.12 / 0.8818	32.24 / 0.8833	32.29 / 0.8861	32.76 / 0.8974	33.23 / 0.9055	**33.62 / 0.9166**
photography		34.77 / 0.9523	38.76 / 0.9762	38.64 / 0.9758	38.93 / 0.9770	39.41 / 0.9787	39.98 / 0.9806	**40.69 / 0.9835**
polyflow		33.03 / 0.9330	39.00 / 0.9670	39.09 / 0.9670	39.42 / 0.9695	40.23 / 0.9744	41.03 / 0.9774	**41.81 / 0.9820**
average		34.50 / 0.9326	38.07 / 0.9609	38.10 / 0.9605	38.34 / 0.9626	38.96 / 0.9664	39.42 / 0.9690	**40.27 / 0.9737**
archpeople		32.51 / 0.9216	35.57 / 0.9513	35.34 / 0.9491	35.55 / 0.9511	35.76 / 0.9531	36.11 / 0.9562	**37.10 / 0.9642**
archwall		36.09 / 0.9118	39.39 / 0.9511	39.54 / 0.9514	39.90 / 0.9558	40.48 / 0.9615	40.66 / 0.9645	**41.66 / 0.9702**
auditorium		25.75 / 0.8174	27.97 / 0.8817	27.49 / 0.8728	28.01 / 0.8833	28.64 / 0.8967	29.25 / 0.9088	**29.94 / 0.9198**
band		29.97 / 0.9030	33.25 / 0.9478	33.02 / 0.9441	33.47 / 0.9496	34.02 / 0.9546	34.26 / 0.9557	**35.11 / 0.9633**
caffe		32.78 / 0.9229	36.77 / 0.9611	36.65 / 0.9599	37.41 / 0.9640	38.47 / 0.9690	38.90 / 0.9714	**39.98 / 0.9750**
camera	x4	41.78 / 0.9848	43.41 / 0.9885	43.62 / 0.9889	43.83 / 0.9892	44.75 / 0.9903	44.82 / 0.9906	**46.64 / 0.9921**
clap		30.75 / 0.9136	34.74 / 0.9529	34.50 / 0.9498	34.93 / 0.9535	35.74 / 0.9592	36.07 / 0.9612	**37.26 / 0.9697**
lake		29.67 / 0.7843	30.48 / 0.8182	30.55 / 0.8185	30.66 / 0.8237	30.93 / 0.8342	31.30 / 0.8436	**31.62 / 0.8575**
photography		32.45 / 0.9217	35.88 / 0.9574	35.76 / 0.9561	35.90 / 0.9575	36.42 / 0.9612	36.83 / 0.9643	**37.80 / 0.9705**
polyflow		31.00 / 0.8990	36.40 / 0.9455	36.39 / 0.9436	36.58 / 0.9475	37.56 / 0.9534	38.00 / 0.9555	**38.72 / 0.9628**
average		32.27 / 0.8980	35.39 / 0.9355	35.29 / 0.9334	35.62 / 0.9375	36.28 / 0.9433	36.62 / 0.9472	**37.58 / 0.9545**

TABLE VI
COMPARISON OF DIFFERENT MODELS. FOR 4× SR, PSNR IS EVALUATED ON VIDEO SEQUENCES FROM [48] AND TESTING TIME IS MEASURED BY GENERATING ONE 1920 × 1080 VIDEO FRAME.

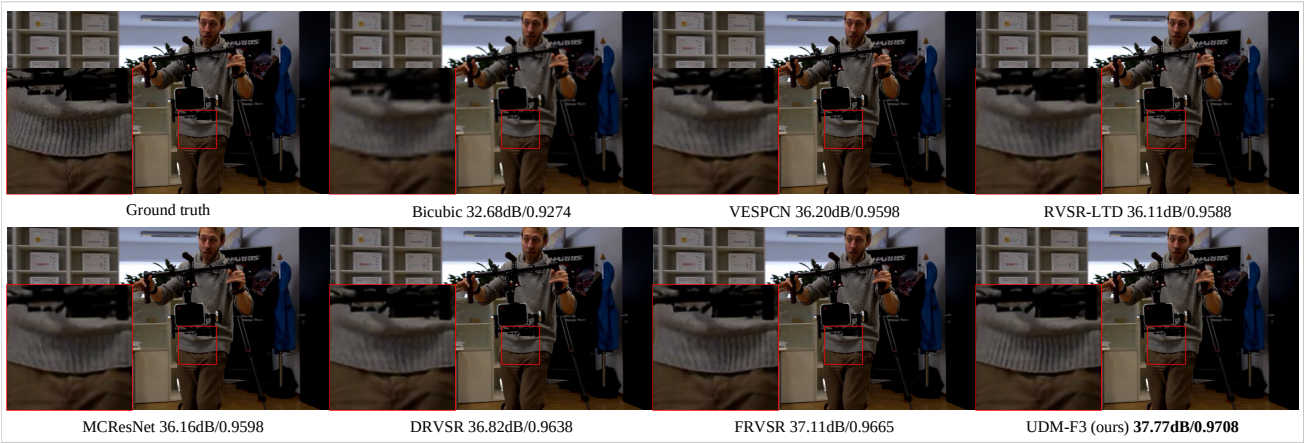
Metric	VESPCN [23]	RVSR-LTD [25]	MCResNet [24]	DRVSR [10]	FRVSR [26]	UDM-F3 (ours)	MTUDM-F5 (ours)
Parameter (M)	0.106	0.374	0.230	1.722	5.057	5.857	5.919
Testing time (s)	0.05	0.06	0.06	0.37	0.18	0.92	1.53
Training time (h)	0.9	1.5	1.3	32.0	78.3	9.0	14.3
PSNR (dB)	35.39	35.29	35.62	36.28	36.62	37.41	37.58

because some video SR approaches need 5 consecutive frames to rebuild the center frame. PSNR and SSIM are calculated by eliminating 8 pixels on each border as in [9], [24]. Note that for 3× magnification, frames of *city* are cut off to 702×576, in order to become an integer multiple of 3. Specific results on Videoset4 testing dataset are demonstrated in Table III. It is seen that the best results are given by our method in all cases.

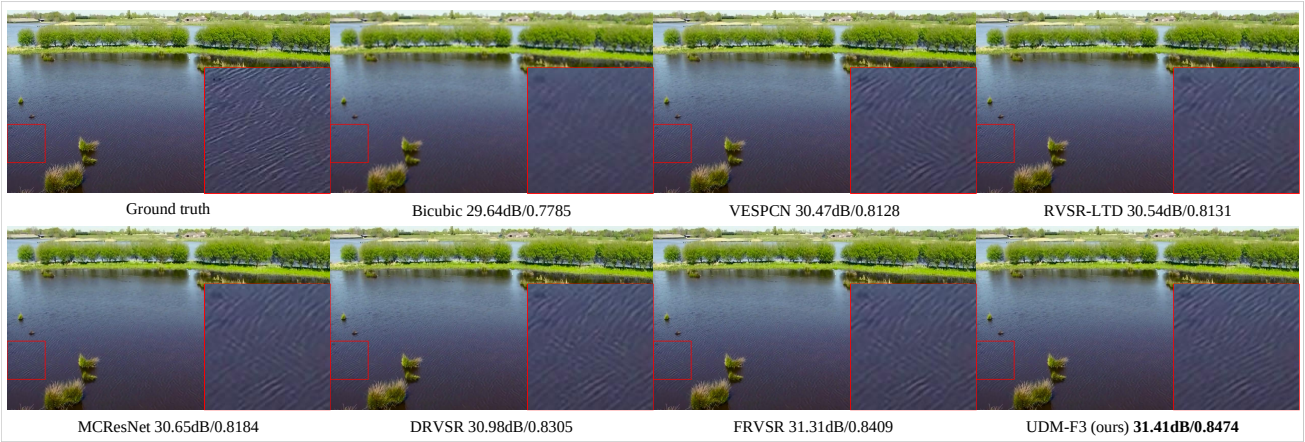
To extensively verify our method, we carry out more experiments on additional datasets. Recently, Liu *et al*. [25] proposed a temporal adaptive neural network that can adaptively determine the optimal scale of temporal dependency for video SR, and conduct experiments on a dataset from [47]. This dataset contains six video sequences: *calendar*, *city*, *foliage*, *penguin*, *temple*, *walk*. Each video sequence has 31 frames, at the resolution of 720 × 576, 704×576, 720×480, 1200×800,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2019.2925844, IEEE Transactions on Circuits and Systems for Video Technology

(a)



(b)



(c)

Fig. 7. Visual Results of different video SR methods. Upscaling factor is 4, and PSNR and SSIM are calculated by cutting off 8 pixels on each border. (a) This frame is from *archpeople*. (b) This frame is from *photography*. (c) This frame is from *lake*.

1200×800 and 720×480 respectively. Deep-DE [47] needs 15 frames forward and 15 frames backward to rebuild the center frame, and there are only 31 frames in each video sequence, so we only compare the center frame. Besides, SSIM values are not reported in [25], thus, we only give the PSNR values. As demonstrated in Table IV, our model MTUDM-F5 achieves the best performance on most video sequences other than *city*. In average, our model performs the best, and surpasses the second best model DRVSR [10] 0.80 dB in PSNR.

More recently, Sajjadi *et al.* [26] proposed a network called FRVSR (frame-recurrent video super-resolution network), which takes the last super-resolved frame to help reconstruct the current input frame. However, they trained FRVSR using a kind of down-sampling scheme rather than Bicubic down-sampling, besides, their code is not available yet. Moreover, codes of VESPCN [23], RVSR-LTD [25]

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2019.2925844, IEEE Transactions on Circuits and Systems for Video Technology

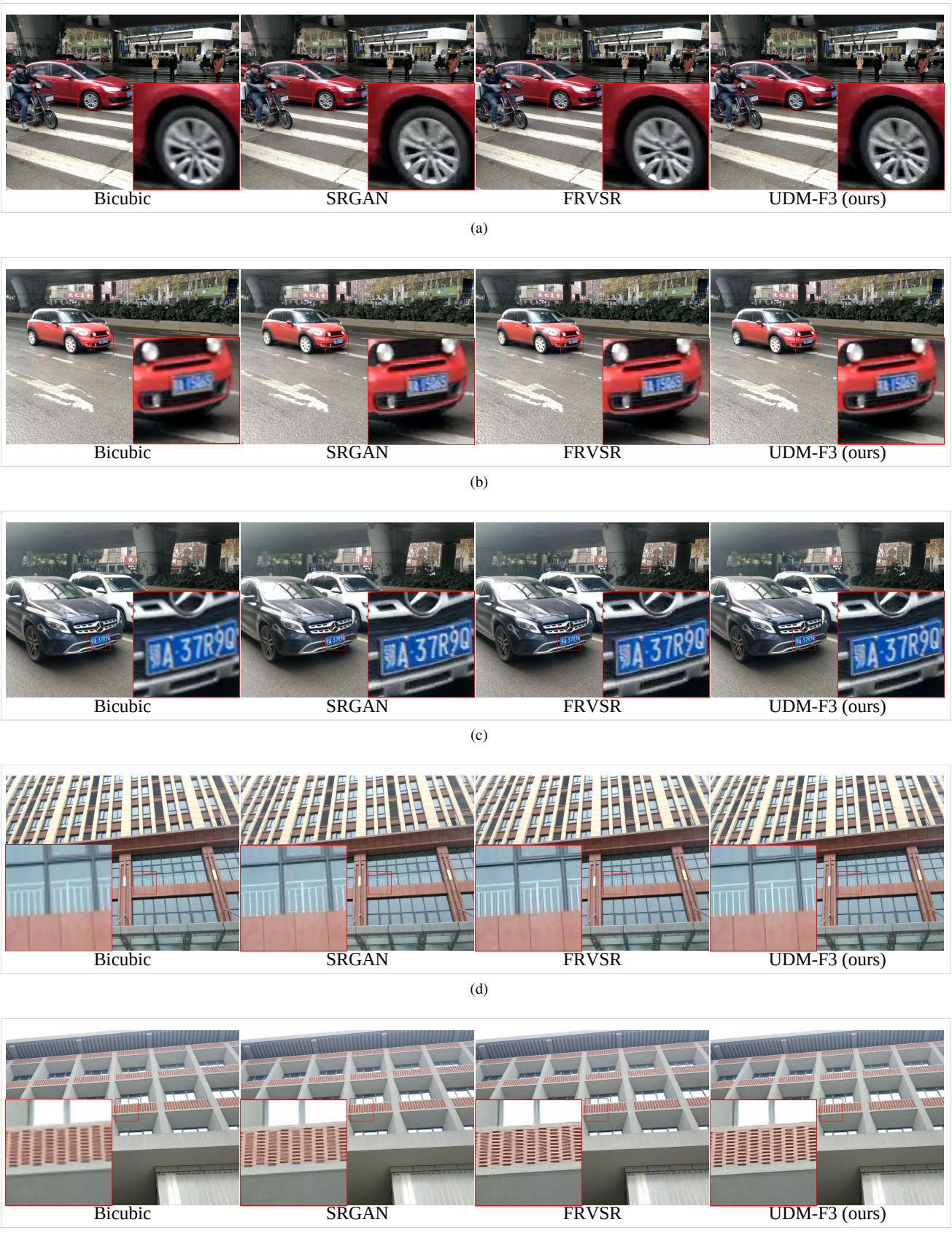


Fig. 8. Visual Results of different video SR methods on real-world LR videos. Upscaling factor is 4, while PSNR and SSIM cannot be calculated without the ground truth image. (a) (b) (c) are about the fast-moving vehicles on the road, it is observed that our model generates clearer results with little blocking artifacts, less noise and better visual quality, compared to other methods. (d) and (e) are about two buildings, and our model produces railings and hollows which are most in line with real-world scenes.

and MCResNet [24] are not available, either. Since various factors (*e.g.* training dataset, platform of code, way of down-sampling) may influence on the performances of different models, we rebuild these models on Tensorflow platform by our training dataset with Bicubic down-sampling scheme. We choose another 10 video sequences as test dataset from [48], which are also adopted by [26]. For convenience, we crop these 10 video sequences (such scenes as archpeople, archwall, auditorium, band, caffe, and so on) to the size of 1272×720, to become an integer multiple of 12. We neglect the first and last two frames, and eliminate 8 pixels on each border to calculate the PSNR and SSIM. As shown in Table V, our model achieves the best results under all conditions (whatever sequences or upscaling factors). Especially, it enjoys more advantages when upscaling factor becomes larger.

Also, we have shown the parameter number, testing time and training time cost of different models in Table VI. FRVSR spends only about 0.19 s for testing, which is rather fast, however, it costs about 78 hours for training due to its special network structure and training strategy. Our models UDM-F3 and MTUDM-F5 take about 0.92 s and 1.53 s for testing, but only require about 9 and 14 hours for training, respectively.

To illustrate different visual effects by comparison methods, we further demonstrate some of the SR images. We show results of our model UDM-F3, which is already able to produce results with good visual quality and runs faster than MTUDM-F5. As shown in Figure 7(a) and Figure 7(b), other models tend to generate blurry stripes, while our model is able to reconstruct stripes with clear details. In Figure 7(c), the water ripples take high-frequency details and are hard to reconstruct. Other video SR models are prone to produce water ripples with wrong directions, while our model generates realistic water ripples with the same directions as the ground truth.

We further conduct experiments on challenging real-world LR videos, which are taken by ourselves (640×480) rather than down-sampled from HR videos. As a representative GAN based SR algorithm, SRGAN [29] has been proposed to generate photo-realistic HR image, whose results show to be lower in PSNR and SSIM but with better visual quality. Thus, we decide to compare our model with SRGAN on real-world videos without calculating the PSNR and SSIM. To show more results, we only compare our model UDM-F3 with Bicubic, SRGAN [29] and FRVSR [26] under $4\times$ SR. As shown in Figure 8(a) and Figure 8(b), it is observed that SRGAN and FRVSR generate results with a lot of blocking artifacts, and Bicubic produces an over-blurred image, while our model generates a clearer frame with little blocking artifacts. As illustrated in Figure 8(c), SRGAN gives an image with a certain amount of noise, and FRVSR produces a deformed license plate, while our model generate a clearer license plate with normal shape. These three videos above are about fast-moving vehicles on the road, which are quite challenging for video SR. Figure 8(d) and Figure 8(e) show the results on two buildings, where SRGAN and FRVSR give twisted iamges with poor visual quality, while our model is able to produce objects like railings and hollows which are most in line with real-world scenes.

VI. Conclusion

In this paper, we have proposed a multi-temporal ultra dense memory (MTUDM) network for video super-resolution. We particularly design an ultra dense memory block (UDMB) to fully exploit the intra-frame spatial correlations and inter-frame temporal correlations, which reveals more realistic image details. By adopting multi-temporal information fusion strategy, we improve the accuracy with only requiring about 62 K extra parameters, which are negligible compared to the video SR model itself (about 1%). We compare our MTUDM model with other recent video SR approaches and demonstrate that our model obtains the state-of-the-art results and surpasses the second-best method about 0.6-1.3 dB in average on extensive public benchmark datasets.

References

[1] M. Shen, P. Xue, and C. Wang, "Down-sampling based video coding using super-resolution technique," *IEEE Transactions on Circuits & Systems for Video Technology*, vol. 21, no. 6, pp. 755–765, 2011.

[2] X. Liu, D. Zhai, R. Chen, X. Ji, D. Zhao, and W. Gao, "Depth super-resolution via joint color-guided internal and external regularizations," *IEEE Trans. Image Process.*, vol. 28, no. 4, pp. 1636–1645, 2019.

[3] L. Zhang, H. Zhang, H. Shen, and P. Li, "A super-resolution reconstruction algorithm for surveillance images," *Signal Processing*, vol. 90, no. 3, pp. 848 – 859, 2010.

[4] W. Dong, F. Fu, G. Shi, X. Cao, J. Wu, G. Li, and X. Li, "Hyperspectral image super-resolution via non-negative structured sparse representation," *IEEE Transactions on Image Processing*, vol. 25, no. 5, pp. 2337–2352, May 2016.

[5] J. Jiang, R. Hu, Z. Wang, Z. Han, and J. Ma, "Facial image hallucination through coupled-layer neighbor embedding," *IEEE Transactions on Circuits & Systems for Video Technology*, vol. 26, no. 9, pp. 1674–1684, 2016.

[6] N. Wang, D. Tao, X. Gao, X. Li, and J. Li, "A comprehensive survey to face hallucination," *International Journal of Computer Vision*, vol. 106, no. 1, pp. 9–30, 2014.

[7] C. Dong, C. L. Chen, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 2, pp. 295–307, 2016.

[8] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *Computer Vision and Pattern Recognition*, 2016, pp. 1646–1654.

[9] A. Kappeler, S. Yoo, Q. Dai, and A. K. Katsaggelos, "Video super-resolution with convolutional neural networks," *IEEE Transactions on Computational Imaging*, vol. 2, no. 2, pp. 109–122, 2016.

[10] X. Tao, H. Gao, R. Liao, J. Wang, and J. Jia, "Detail-revealing deep video super-resolution," in *IEEE International Conference on Computer Vision*, 2017, pp. 4482–4490.

[11] Z. Wang, P. Yi, K. Jiang, J. Jiang, Z. Han, T. Lu, and J. Ma, "Multi-memory convolutional neural network for video super-resolution," *IEEE Transactions on Image Processing*, vol. 28, no. 5, pp. 2530–2544, 2019.

[12] L. Zhou, Z. Wang, Y. Luo, and Z. Xiong, "Separability and compactness network for image recognition and superresolution," *IEEE Transactions on Neural Networks and Learning Systems*, 2019.

[13] L. Zhang and X. Wu, "An edge-guided image interpolation algorithm via directional filtering and data fusion," *IEEE Transactions on Image Processing*, vol. 15, no. 8, pp. 2226–2238, 2006.

[14] C. E. Duchon, "Lanczos filtering in one and two dimensions," *Journal of Applied Meteorology*, vol. 18, no. 8, pp. 1016–1022, 1979.

[15] X. Liu, D. Zhao, R. Xiong, S. Ma, W. Gao, and H. Sun, "Image interpolation via regularized local linear regression," *IEEE Transactions on Image Processing*, vol. 20, no. 12, pp. 3455–3469, Dec 2011.

[16] X. Liu, D. Zhai, R. Chen, X. Ji, D. Zhao, and W. Gao, "Depth restoration from rgb-d data via joint adaptive regularization and thresholding on manifolds," *IEEE Trans. Image Process.*, vol. 28, no. 3, pp. 1068–1079, 2019.

[17] S. Farsiu, M. Robinson, M. Elad, and P. Milanfar, "Fast and robust multiframe super resolution," *IEEE Transactions on Image Processing*, vol. 13, no. 10, pp. 1327–1344, 2004.

[18] C. Liu and D. Sun, "On bayesian adaptive video super resolution." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 2, pp. 346–60, 2014.
[19] Z. Ma, R. Liao, X. Tao, L. Xu, J. Jia, and E. Wu, "Handling motion blur in multi-frame super-resolution," in *Computer Vision and Pattern Recognition*, 2015, pp. 5224–5232.
[20] C. Wang, P. Xue, and W. Lin, "Improved super-resolution reconstruction from video," *IEEE Transactions on Circuits & Systems for Video Technology*, vol. 16, no. 11, pp. 1411–1422, 2006.
[21] J. Chen, J. L. Nunez-Yanez, and A. Achim, "Bayesian video super-resolution with heavy-tailed prior models," *IEEE Transactions on Circuits & Systems for Video Technology*, vol. 24, no. 6, pp. 905–914, 2014.
[22] Y. Huang, W. Wang, and L. Wang, "Bidirectional recurrent convolutional networks for multi-frame super-resolution," in *International Conference on Neural Information Processing Systems*, 2015, pp. 235–243.
[23] J. Caballero, C. Ledig, A. Aitken, A. Acosta, J. Totz, Z. Wang, and W. Shi, "Real-time video super-resolution with spatio-temporal networks and motion compensation," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 2848–2857.
[24] D. Li and Z. Wang, "Video superresolution via motion compensation and deep residual learning," *IEEE Transactions on Computational Imaging*, vol. 3, no. 4, pp. 749–762, 2017.
[25] D. Liu, Z. Wang, Y. Fan, X. Liu, Z. Wang, S. Chang, and T. Huang, "Robust video super-resolution with learned temporal dynamics," in *IEEE International Conference on Computer Vision*, 2017, pp. 2526–2534.
[26] M. S. M. Sajjadi, R. Vemulapalli, and M. Brown, "Frame-recurrent video super-resolution," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018, pp. 6626–6634.
[27] X. Shi, Z. Chen, H. Wang, W. C. Woo, W. C. Woo, and W. C. Woo, "Convolutional lstm network: a machine learning approach for precipitation nowcasting," in *International Conference on Neural Information Processing Systems*, 2015, pp. 802–810.
[28] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang, "Deep laplacian pyramid networks for fast and accurate super-resolution," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5835–5843.
[29] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 105–114.
[30] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Computer Vision and Pattern Recognition*, 2016, pp. 1874–1883.
[31] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced deep residual networks for single image super-resolution," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 1132–1140.
[32] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018, pp. 2472–2481.
[33] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Training very deep networks," in *Advances in Neural Information Processing Systems 28*, 2015, pp. 2377–2385.
[34] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.
[35] G. Huang, Z. Liu, L. V. D. Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2261–2269.
[36] G. Huang, D. Chen, T. Li, F. Wu, L. van der Maaten, and K. Q. Weinberger, "Multi-scale dense convolutional networks for efficient prediction," *CoRR*, vol. abs/1703.09844, 2017. [Online]. Available: http://arxiv.org/abs/1703.09844
[37] A. Dosovitskiy, P. Fischery, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. V. D. Smagt, D. Cremers, and T. Brox, "Flownet: Learning optical flow with convolutional networks," in *IEEE International Conference on Computer Vision*, 2015, pp. 2758–2766.
[38] Sampepose, "Flownets," https://github.com/sampepose/flownet2-tf, 2017, last accessed on 2018-11-22.
[39] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "Flownet 2.0: Evolution of optical flow estimation with deep networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1647–1655.
[40] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial transformer networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 2017–2025.
[41] J. Shuiwang, Y. Ming, X. Wei, and Y. Kai, "3d convolutional neural networks for human action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 221–231, 2012.
[42] R. Timofte, K. M. Lee, X. Wang, Y. Tian, K. Yu, Y. Zhang, S. Wu, C. Dong, L. Lin, and Y. Qiao, "Ntire 2017 challenge on single image super-resolution: Methods and results," in *Computer Vision and Pattern Recognition Workshops*, 2017, pp. 1110–1121.
[43] A. Bruhn, J. Weickert, and C. Schnrr, "Lucas/kanade meets horn/schunck: Combining local and global optic flow methods," *International Journal of Computer Vision*, vol. 61, no. 3, pp. 211–231, 2005.
[44] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *IEEE International Conference on Computer Vision*, 2016, pp. 1026–1034.
[45] R. Timofte, V. D. Smet, and L. V. Gool, "A+: Adjusted anchored neighborhood regression for fast super-resolution," in *Asian Conference on Computer Vision*, 2014, pp. 111–126.
[46] J. Kim, J. K. Lee, and K. M. Lee, "Deeply-recursive convolutional network for image super-resolution," in *Computer Vision and Pattern Recognition*, 2016, pp. 1637–1645.
[47] R. Liao, X. Tao, R. Li, Z. Ma, and J. Jia, "Video super-resolution via deep draft-ensemble learning," in *IEEE International Conference on Computer Vision*, 2015, pp. 531–539.
[48] Vimeo, "Vimeo," http://www.vimeo.com, 2018, last accessed on 2018-11-22.
[49] M. Bevilacqua, A. Roumy, C. Guillemot, and M. line Alberi Morel, "Low-complexity single-image super-resolution based on nonnegative neighbor embedding," in *Proceedings of the British Machine Vision Conference*, 2012, pp. 135.1–135.10.
[50] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," in *International Conference on Curves and Surfaces*, 2010, pp. 711–730.
[51] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 5, pp. 898–916, 2011.
[52] J. B. Huang, A. Singh, and N. Ahuja, "Single image super-resolution from transformed self-exemplars," in *Computer Vision and Pattern Recognition*, 2015, pp. 5197–5206.
[53] Y. Matsui, K. Ito, Y. Aramaki, A. Fujimoto, T. Ogawa, T. Yamasaki, and K. Aizawa, "Sketch-based manga retrieval using manga109 dataset," *Multimedia Tools and Applications*, vol. 76, no. 20, pp. 21 811–21 838, 2017.
[54] J. Caballero, "Videoset4," https://twitter.box.com/v/vespcn-vid4, 2016, last accessed on 2018-11-22.

Peng Yi (M'17) received the B.S. degree in Faculty of Electronic Information and Electrical Engineering from Dalian University of Technology, Dalian, China, in 2017.

He is currently working toward the Ph.D. degree under the supervision of Prof. Zhongyuan Wang in the School of Computer, Wuhan University.

Zhongyuan Wang (M'13) received the Ph.D. degree in communication and information system from Wuhan University, Wuhan, China, in 2008.

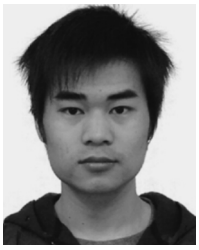
He is currently a Professor with the School of Computer, Wuhan University. He is also directing three projects funded by the National Natural Science Foundation Program of China.

Kui Jiang (M'16) received the B.S. degree in Xinjiang University, China, in 2017.

He is currently working toward the MA degree under the supervision of Prof. Zhongyuan Wang in the School of Computer, Wuhan University.

Zhenfeng Shao (M'16) received the Ph.D. degree from Wuhan University,Wuhan, China, in 2004.

He is currently a Professor with the State Key Laboratory of Information Engineering in Surveying, Mapping, and Remote Sensing, Wuhan University. His research interest focuses on computer vision.



Jiayi Ma (M'16) received the B.S. degree in Information and Computing Science and the Ph.D. degree in Control Science and Engineering, both from the Huazhong University of Science and Technology, Wuhan, China, in 2008 and 2014, respectively.

From 2012 to 2013, he was an Exchange Student with the Department of Statistics, University of California at Los Angeles, Los Angeles, CA, USA. He is currently an Associate Professor with the Electronic Information School, Wuhan University, Wuhan, China, where he was a Postdoctoral Researcher from 2014 to 2015. He has authored or co-authored over 100 refereed journal and conference papers, including IEEE TPAMI/TIP/TSP/TNNLS/TGRS/TCYB/TMM/TCSVT, IJCV, CVPR, IJCAI, AAAI, ICRA, IROS, ACM MM. He has won the Natural Science Award of Hubei Province (first class) as the first author. He has received the CAAI (Chinese Association for Artificial Intelligence) Excellent Doctoral Dissertation Award (a total of 8 winners in China), and the CAA (Chinese Association of Automation) Excellent Doctoral Dissertation Award (a total of 10 winners in China). His current research interests include the areas of computer vision, machine learning, and pattern recognition.