

ColorMNet: A Memory-based Deep Spatial-Temporal Feature Propagation Network for Video Colorization

Yixin Yang, Jiangxin Dong, Jinhui Tang, and Jinshan Pan

Nanjing University of Science and Technology

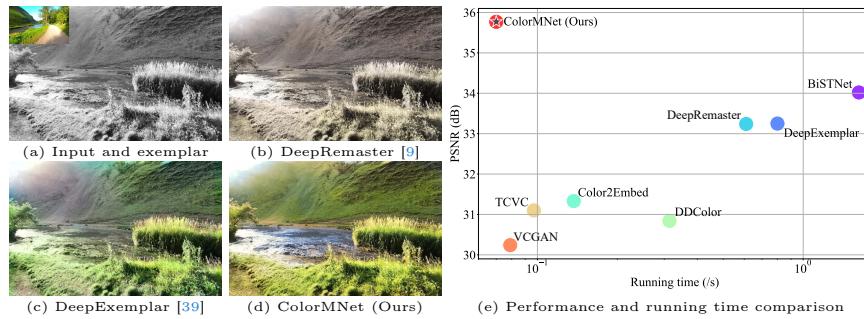


Fig. 1: Colorization results on a real-world video and model performance comparisons between our proposed ColorMNet and other methods on the DAVIS [25] dataset in terms of PSNR and running time. State-of-the-art methods [9, 39] do not generate well-colorized images in (b) and (c). In contrast, by exploring the features from large-pretrained visual models to estimate robust spatial features for each frame, effectively propagating these features along the temporal dimension based on memory mechanisms for far-apart frames, and exploiting the video property that adjacent frames contain similar contents, our method accurately restores the colors on the grass and generates a realistic image in (d). (e) shows that the proposed ColorMNet performs favorably against state-of-the-art methods in terms of accuracy and running time. The size of the test images for measuring the running time is 960×536 pixels.

Abstract. How to effectively explore spatial-temporal features is important for video colorization. Instead of stacking multiple frames along the temporal dimension or recurrently propagating estimated features that will accumulate errors or cannot explore information from far-apart frames, we develop a memory-based feature propagation module that can establish reliable connections with features from far-apart frames and alleviate the influence of inaccurately estimated features. To extract better features from each frame for the above-mentioned feature propagation, we explore the features from large-pretrained visual models to guide the feature estimation of each frame so that the estimated features can model complex scenarios. In addition, we note that adjacent frames usually contain similar contents. To explore this property for better spatial and temporal feature utilization, we develop a local attention module to aggregate the features from adjacent frames in a spatial-temporal neighborhood. We formulate our memory-based feature

propagation module, large-pretrained visual model guided feature estimation module, and local attention module into an end-to-end trainable network (named ColorMNet) and show that it performs favorably against state-of-the-art methods on both the benchmark datasets and real-world scenarios. The source code and pre-trained models will be available at <https://github.com/yyang181/colormnet>.

Keywords: Exemplar-based video colorization · Deep convolutional neural network · Feature propagation

1 Introduction

Due to the technical limitations of old imaging devices, lots of videos captured in the last century are in black and white, making them less visually appealing on modern display devices. As most of these videos have historical values and are difficult to reproduce, it is of great need to colorize them.

Restoring high-quality colorized videos is challenging as it not only needs to handle the colorization of each frame but also requires exploring temporal information from the video sequences. Therefore, directly applying existing image colorization methods [3, 10, 12, 16, 19, 29, 35, 40, 42] does not generate satisfactory colorized videos as minor perturbations in consecutive input video frames may lead to substantial differences in colorized video results. To overcome this, numerous methods model the temporal information from inter-frames by stacking multiple frames along the temporal dimension [1, 9] or recurrently propagating features [17, 32, 33, 39]. Although these approaches show better performance than the ones based on single image colorization, stacking multiple frames along the temporal dimension cannot effectively leverage spatial-temporal prior from adjacent frames and requires a large amount of GPU memory. In addition, recurrent-based feature propagation is not able to effectively explore long-range information, leading to unsatisfactory results for frames far apart.

To better explore long-range temporal information, several approaches [20, 36] develop bidirectional recurrent-based feature propagation methods for video colorization. As the recurrent-based feature propagation treats the features of each frame equally, if the features are not estimated accurately, the errors will accumulate, thus affecting the final video colorization. Therefore, it is still challenging to effectively model temporal information from long-range frames.

In addition to the temporal information exploration, how to extract good features from each frame plays a significant role in video colorization. Existing methods [1, 17, 32, 36, 39, 44] usually utilize a pretrained VGG [13] or ResNet-101 [6] to extract features from each frame. These methods are able to model local structures but are less effective for exploiting non-local and semantic structures, *e.g.*, complex scenes with multiple objects. To restore high-quality videos, it is of great interest to develop a better feature representation method that is able to characterize the non-local and semantic properties of each frame.

In this paper, we present a memory-based deep spatial-temporal feature propagation network for video colorization. Note that the robustness of the spa-

tial features extracted from each frame is important, we first develop a large-pretrained visual model guided feature estimation (PVGFE) module, which is motivated by the success of large-pretrained visual models [22] in generating robust visual features to facilitate the spatial feature estimation. However, simply recurrently propagating the estimated spatial features or directly stacking them along the temporal dimension does not effectively explore temporal information for video colorization. Moreover, it requires a large amount of GPU memory capacity to store past frame representations when the spatial resolution is large and videos are long. To overcome these problems, we then propose a memory-based feature propagation (MFP) module that can not only adaptively explore and propagate useful features from far-apart frames but also reduce memory consumption. In addition, we note that adjacent frames of a video usually contain similar contents and thus develop a local attention (LA) module to better utilize the spatial and temporal features. Taken together, the memory-based deep spatial-temporal feature propagation network, called ColorMNet, is able to generate high-quality video colorization results (see Figure 1(e)).

The main contributions are summarized as follows:

- We propose a large-pretrained visual model guided feature estimation module to model non-local and semantic structures of each frame for colorization.
- We develop a memory-based feature propagation module to adaptively explore temporal features from far-apart frames and reduce memory usage.
- We develop a local attention module to explore similar contents of adjacent frames for better video colorization.
- We formulate the proposed network into an end-to-end trainable framework and show that it performs favorably against state-of-the-art methods on both the benchmark datasets and real-world scenarios.

2 Related Work

User-guided image colorization. Since the colorization problem is ill-posed, conventional image colorization methods usually adopt local user hints [2, 7, 19, 21, 26, 27, 37, 42, 43] to make this problem well-posed. However, these methods do not fully exploit the property of video sequences and usually need to solve temporal consistency problems. In addition, their colorization performance for individual frame is usually far from satisfactory as estimating global and semantic features is challenging.

Automatic video colorization. Instead of using user-guided image methods, several approaches explore deep learning to solve video colorization. In [17, 44], both the colorization performance and the temporal consistency are enhanced by recurrently propagating features from adjacent frames for video colorization. In [20], Liu *et al.* use bidirectional propagation to explore temporal features and introduces a self-regularization learning scheme to minimize the prediction difference obtained with different time steps. Although using feature propagation improves the temporal consistency, it is not a trivial task to estimate the spatial features from each frame due to the inherent complexity of scenes in videos. Moreover, these automatic video colorization methods [24, 28, 35] may work well on synthetic datasets but lack generality on diverse real-world scenarios.

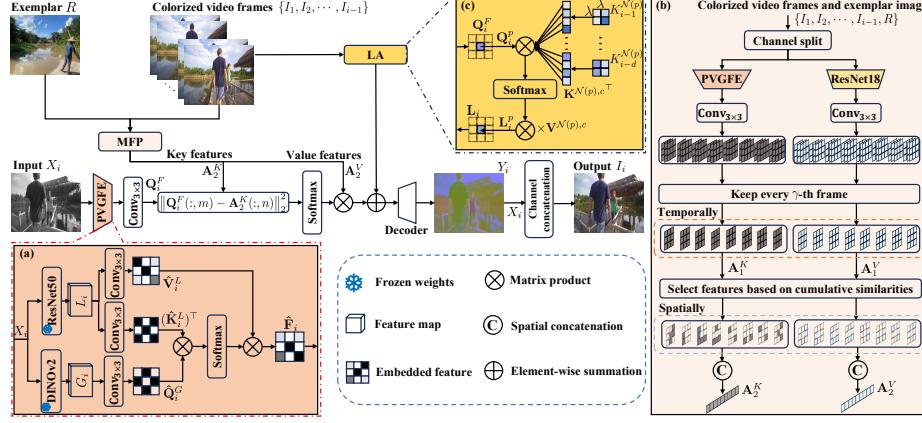


Fig. 2: An overview of the proposed ColorMNet. The core components of our method include: (a) large-pretrained visual model guided feature estimation (PVGFE) module, (b) memory-based feature propagation (MFP) module and (c) local attention (LA).

Exemplar-based video colorization. Exemplar-based methods aim to generate videos that are faithful to the exemplar with enhanced temporal consistency. In [39], Zhang *et al.* employ a recurrent-based feature propagation module to explore temporal features for latent frame restoration. In [9], Iizuka *et al.* propose stacking multiple frames along the temporal dimension to obtain better performance. To better explore spatial and temporal features, Chen *et al.* [1] adopt ResNets [6] instead of the commonly-used VGG [13] model for better spatial feature estimation and split video sequences into frame blocks for long-term spatiotemporal dependency. In [36], Yang *et al.* use bidirectional propagation to gradually propagate features and use optical flow models [31] to align adjacent frames. Recurrently propagating information from long-range frames or stacking multiple frames improves the colorization performance. However, if there are inaccurately estimated features of long-range frames, the errors will be accumulated, which thus affects video colorization. Additionally, when dealing with long videos, these methods often consume significant GPU memory and suffer from slow inference speeds, posing substantial challenges to video colorization.

3 ColorMNet

Our goal is to develop an effective and efficient video colorization method to restore high-quality videos with low GPU memory requirements. The proposed ColorMNet contains a large-pretrained visual model guided feature estimation (PVGFE) module to extract spatial features from each frame, a memory-based feature propagation (MFP) module that is able to adaptively explore the temporal features from far-apart frames, and a local attention (LA) module that is used to explore the similar contents from adjacent frames for better spatial and temporal feature utilization. Figure 2 illustrates the overview of the proposed method. In the following, we explain each module in detail.

3.1 PVGFE module

To estimate robust spatial features, we explore the features learned from large-pretrained visual models as they are able to model the non-local and semantic information and are robust to numerous scenarios. Note that one of the large-pretrained visual models, *i.e.*, DINOv2 [22], adopts ViT [4] as the main feature extractor and generates all-purpose visual features facilitating both image-level (classification) and pixel level (segmentation) tasks due to its robust feature representation ability. In this paper, we utilize the global features learned from DINOv2 to guide the local features learned from CNNs for better feature estimation of each frame.

Given the input grayscale frames $\{X_i\}_{i=1}^N$ (where $X_i \in \mathbb{R}^{H \times W \times 1}$, $H \times W$ denotes the spatial resolution, N is the number of frames of the input video), we first extract features $\{G_i\}_{i=1}^N$ and $\{L_i\}_{i=1}^N$ by applying the pretrained DINOv2 and ResNet50 [6] to $\{X_i\}_{i=1}^N$, respectively. Then we use the cross attention [38] to fuse $\{G_i\}_{i=1}^N$ and $\{L_i\}_{i=1}^N$ for obtaining robust spatial features.

Specifically, we first extract the query feature Q_i^G from G_i , the key feature K_i^L and the value feature V_i^L from L_i by:

$$Q_i^G = \text{Conv}_{3 \times 3}(G_i), \quad (1a)$$

$$K_i^L = \text{Conv}_{3 \times 3}(L_i), \quad (1b)$$

$$V_i^L = \text{Conv}_{3 \times 3}(L_i), \quad (1c)$$

where $\{Q_i^G, K_i^L, V_i^L\} \in \mathbb{R}^{\hat{H} \times \hat{W} \times \hat{C}}$, $\hat{H} \times \hat{W}$ and \hat{C} denote the spatial and channel dimensions, respectively; $\text{Conv}_{3 \times 3}(\cdot)$ denotes a convolution with the filter size of 3×3 pixels. We denote the matrix forms of Q_i^G , K_i^L , V_i^L as $\hat{\mathbf{Q}}_i^G$, $\hat{\mathbf{K}}_i^L$, $\hat{\mathbf{V}}_i^L$, and obtain the fused feature by:

$$\hat{\mathbf{F}}_i = \text{softmax} \left(\frac{\hat{\mathbf{Q}}_i^G (\hat{\mathbf{K}}_i^L)^\top}{\alpha} \right) \hat{\mathbf{V}}_i^L, \quad (2)$$

where $\hat{\mathbf{Q}}_i^G \in \mathbb{R}^{\hat{C} \times \hat{H}\hat{W}}$, $\hat{\mathbf{K}}_i^L \in \mathbb{R}^{\hat{C} \times \hat{H}\hat{W}}$ and $\hat{\mathbf{V}}_i^L \in \mathbb{R}^{\hat{C} \times \hat{H}\hat{W}}$ are obtained by reshaping tensors [38] from the original size $\mathbb{R}^{\hat{H} \times \hat{W} \times \hat{C}}$; $\text{softmax}(\cdot)$ denotes the softmax operation that is applied to each row of the matrix; α is a scaling factor.

In our implementation, we take the features of the last 4 layers of ViT-S/14 from DINOv2 and concatenate them together in channel dimension as the feature G_i . For the feature L_i , we take stage-4 features with stride 16 from the base ResNet50 [6]. Finally, the feature $\hat{\mathbf{F}}_i \in \mathbb{R}^{\hat{C} \times \hat{H}\hat{W}}$ is reshaped into $F_i \in \mathbb{R}^{\hat{H} \times \hat{W} \times \hat{C}}$ for the following processing.

3.2 MFP module

Inspired by the efficient mechanisms of human brains in memorizing long-term information, *i.e.*, paying more attention to frequently used information, we propose a memory-based feature propagation module to effectively and efficiently explore temporal features by establishing reliable connections with features from far-apart frames and alleviating the influence of inaccurately estimated features.

Assuming that we have predicted $i - 1$ color frames $\{I_1, \dots, I_{i-1}\}$ with their chrominance channels $\{Y_1, \dots, Y_{i-1}\}$ and luminance channels $\{X_1, \dots, X_{i-1}\}$

(*i.e.*, the input grayscale frames), we aim to estimate the chrominance channel Y_i of the i -th color frame I_i based on $\{Y_1, \dots, Y_{i-1}\}$ and the given exemplar R .

First, we extract features $\{F_1, F_2, \dots, F_i\}$ and F_r from $\{X_1, X_2, \dots, X_i\}$ and the luminance channel X_r of R using the proposed PVGFE module for global and semantic features in the spatial dimension. Meanwhile, we extract features $\{E_1, \dots, E_{i-1}\}$ and E_r from $\{Y_1, \dots, Y_{i-1}\}$ and the chrominance channel Y_r of R using a lightweight pretrained ResNet18 [6]. Then, we generate the embedded query, key, and value features by:

$$Q_i^F = \text{Conv}_{3 \times 3}(F_i), \quad (3a)$$

$$\{K_j^F\}_{j=1}^{i-1} = \{\text{Conv}_{3 \times 3}(F_j)\}_{j=1}^{i-1}, \quad (3b)$$

$$K_r^F = \text{Conv}_{3 \times 3}(F_r), \quad (3c)$$

$$\{V_j^E\}_{j=1}^{i-1} = \{\text{Conv}_{3 \times 3}(E_j)\}_{j=1}^{i-1}, \quad (3d)$$

$$V_r^E = \text{Conv}_{3 \times 3}(E_r). \quad (3e)$$

As $\{K_1^F, \dots, K_{i-1}^F\}$ and $\{V_1^E, \dots, V_{i-1}^E\}$ contain valuable historical information of previously colorized video frames $\{I_1, \dots, I_{i-1}\}$, they facilitate the establishment of long-range temporal correspondences between the contents of the current frame and those of the previously colorized video frames. However, memorizing all of the historical frames results in significant GPU memory consumption, especially as the number of colorized frames increases. As a trade-off between long-range correspondence and memory consumption, we keep every γ frame and discard the remaining ones that contain similar contents to obtain more temporally compact and representative features.

In particular, given that adjacent frames are mutually redundant, we first merge the temporal features by concatenating every γ frame, thereby establishing reliable connections with far-apart frames under constrained memory consumption, and obtain the aggregated key features \mathbf{A}_1^K by:

$$\mathbf{A}_1^K = \text{Concat}(\mathbf{K}_\gamma^F, \mathbf{K}_{2\gamma}^F, \dots, \mathbf{K}_{z\gamma}^F), \quad (4)$$

where $\{\mathbf{K}_\gamma^F, \mathbf{K}_{2\gamma}^F, \dots, \mathbf{K}_{z\gamma}^F\}$ are the matrix forms of $\{K_\gamma^F, K_{2\gamma}^F, \dots, K_{z\gamma}^F\}$, $z = \lfloor \frac{i-1}{\gamma} \rfloor$, $\lfloor \cdot \rfloor$ denotes the round down operation, and $\text{Concat}(\cdot)$ denotes the spatial dimension concatenation operation.

Note that errors are likely to accumulate when the features of some frames are not estimated accurately. In addition, the high dimension ($\hat{H}\hat{W}$) of \mathbf{A}_1^K makes it impossible to handle lots of video frames with limited GPU memory capacity. To overcome these problems, we then aggregate \mathbf{A}_1^K into a more spatially compact and less error-prone form by selecting better features with higher usage. However, the computation of the feature usage frequency relies on a sufficient number of colorized frames. Therefore, when the number of colorized frames is small (*i.e.*, $z < N_s$), we directly use \mathbf{A}_2^K as the output of our proposed MFP module, which is defined as:

$$\mathbf{A}_2^K = \text{Concat}(\mathbf{K}_r^F, \mathbf{A}_1^K), \quad (5)$$

where \mathbf{K}_r^F is the matrix form of K_r^F . When a sufficient number of frames is colorized (*i.e.*, $z = N_s$), we aggregate \mathbf{A}_1^K by compressing the earlier N_e frames to reduce GPU memory consumption and alleviate the influence of inaccurately

estimated features for better video colorization. After the aggregation, the total number of aggregated frames reduced from $z = N_s$ to $z = N_s - N_e$ and we use \mathbf{A}_2^K in (5) as the output of the MFP module until z reaches N_s again. In the following, we explain the situation when $z = N_s$ in detail.

We first define the cumulative similarities $\{\mathbf{S}_\gamma, \dots, \mathbf{S}_{z\gamma}\}$ for the key features $\{\mathbf{K}_\gamma^F, \dots, \mathbf{K}_{z\gamma}^F\}$, which are aggregated in (4), as:

$$\mathbf{S}_\gamma = \sum_{j=\gamma+1}^{i-1} \sum_{m=1}^{\hat{H}\hat{W}} (\text{softmax}(-\mathbf{C}^j)), \mathbf{C}^j = (\mathbf{C}_{m,n}^j), \quad (6a)$$

$$\mathbf{C}_{m,n}^j = \|\mathbf{K}_j^F(:, m) - \mathbf{K}_\gamma^F(:, n)\|_2^2, \quad (6b)$$

where $\mathbf{S}_\gamma \in \mathbb{R}^{1 \times \hat{H}\hat{W}}$ and $\mathbf{C}^j \in \mathbb{R}^{\hat{H}\hat{W} \times \hat{H}\hat{W}}$, $\mathbf{K}_j^F(:, m)$ denotes the m -th feature vector in \mathbf{K}_j^F and $\mathbf{K}_\gamma^F(:, n)$ denotes the n -th feature vector in \mathbf{K}_γ^F , $\{\mathbf{K}_j^F, \mathbf{K}_\gamma^F\} \in \mathbb{R}^{\hat{C}^k \times \hat{H}\hat{W}}$, $\|\cdot\|_2$ denotes the Euclidean distance calculation. Then, we normalize \mathbf{S}_γ by dividing it by the number of frames in $\{\mathbf{K}_j^F\}_{j=\gamma+1}^{i-1}$ for fairness consideration and obtain $\mathbf{S}'_\gamma = \frac{\mathbf{S}_\gamma}{i-1-\gamma}$, which can be utilized to estimate the probability that the feature \mathbf{K}_γ^F is accurately predicted as \mathbf{S}'_γ describes how frequently the feature is used. We further define a top- M operation $\mathcal{T}_M(\cdot)$ to select the best M pixels of all features from the earlier N_e frames $\{\mathbf{K}_\gamma^F, \dots, \mathbf{K}_{N_e\gamma}^F\}$, based on the highest M values in $\{\mathbf{S}'_\gamma, \dots, \mathbf{S}'_{N_e\gamma}\}$ as:

$$\mathbf{K}^c = \text{Concat}(\mathbf{K}_\gamma^F, \dots, \mathbf{K}_{N_e\gamma}^F), \quad (7a)$$

$$\mathbf{S}^c = \text{Concat}(\mathbf{S}'_\gamma, \dots, \mathbf{S}'_{N_e\gamma}), \quad (7b)$$

$$\mathcal{T}_M(\mathbf{K}^c) = \text{Concat}(\mathbf{K}^c(:, 1), \dots, \mathbf{K}^c(:, M)), \quad (7c)$$

where $\mathbf{K}^c \in \mathbb{R}^{\hat{C}^k \times N_e \hat{H}\hat{W}}$, $\mathbf{S}^c \in \mathbb{R}^{1 \times N_e \hat{H}\hat{W}}$, and $\{\mathbf{K}^c(:, 1), \dots, \mathbf{K}^c(:, M)\}$ denote the M feature vectors in \mathbf{K}^c that satisfy $\{\mathbf{S}^c(:, 1), \dots, \mathbf{S}^c(:, M)\}$ are the top- M values in \mathbf{S}^c . Then we obtain the aggregated key features \mathbf{A}_2^K by:

$$\mathbf{A}_2^K = \text{Concat}(\mathcal{T}_M(\mathbf{K}^c), \mathbf{K}_r^F, \mathbf{K}_{(N_e+1)\gamma}^F, \dots, \mathbf{K}_{z\gamma}^F), \quad (8)$$

where $\mathbf{A}_2^K \in \mathbb{R}^{\hat{C}^k \times T}$, $T = M + (1 + z - N_e) \hat{H}\hat{W}$. Similarly, we obtain the aggregated value features $\mathbf{A}_2^V \in \mathbb{R}^{\hat{C}^v \times T}$ by aggregating $\{V_r^E, \{V_j^E\}_{j=1}^{i-1}\}$.

To fully exploit the temporal information contained in \mathbf{A}_2^K and \mathbf{A}_2^V , we use a commonly used L_2 similarity to find the features in \mathbf{A}_2^K that are most similar to \mathbf{Q}_i^F , where $\mathbf{Q}_i^F \in \mathbb{R}^{\hat{C}^k \times \hat{H}\hat{W}}$ is the matrix form of Q_i^F , by:

$$\mathbf{W}_i = \text{softmax}(-\mathbf{D}^i), \mathbf{D}^i = (\mathbf{D}_{m,n}^i), \quad (9a)$$

$$\mathbf{D}_{m,n}^i = \|\mathbf{Q}_i^F(:, m) - \mathbf{A}_2^K(:, n)\|_2^2, \quad (9b)$$

where $\{\mathbf{W}_i, \mathbf{D}^i\} \in \mathbb{R}^{\hat{H}\hat{W} \times T}$, $\mathbf{Q}_i^F(:, m)$ denotes the m -th feature vector in \mathbf{Q}_i^F and $\mathbf{A}_2^K(:, n)$ denotes the n -th feature vector in \mathbf{A}_2^K . Then, we reconstruct the i -th estimated feature $\mathbf{V}_i \in \mathbb{R}^{\hat{C}^v \times \hat{H}\hat{W}}$ of the chrominance channel by mapping the aggregated value features \mathbf{A}_2^V based on \mathbf{W}_i as:

$$\mathbf{V}_i = \mathbf{A}_2^V(\mathbf{W}_i)^\top. \quad (10)$$

Finally, we obtain the estimated feature V_i by reshaping \mathbf{V}_i to its original size $\mathbb{R}^{\hat{H} \times \hat{W} \times \hat{C}^v}$. In the following, we enhance V_i by a local attention (LA) module.

3.3 LA module

As adjacent frames contain similar contents that may be useful to complement the long-range information captured by MFP, we develop a local attention (LA) module to explore better spatial-temporal features.

We first use $Q_i^p \in \mathbb{R}^{1 \times 1 \times \hat{C}^k}$ to represent the feature Q_i^F in (3a) at the spatial location $p \in \mathbb{R}^{\hat{H} \times \hat{W}}$. Next, we formulate the past d key features in (3b) and past d value features in (3d) of the spatial-temporal neighborhood corresponding to Q_i^p as:

$$K^{\mathcal{N}(p),c} = \text{Concat}(K_{i-d}^{\mathcal{N}(p)}, \dots, K_{i-1}^{\mathcal{N}(p)}), \quad (11a)$$

$$V^{\mathcal{N}(p),c} = \text{Concat}(V_{i-d}^{\mathcal{N}(p)}, \dots, V_{i-1}^{\mathcal{N}(p)}), \quad (11b)$$

where $K_j^{\mathcal{N}(p)}$ and $V_j^{\mathcal{N}(p)}$ denote the j -th key feature and value feature corresponding to a $\lambda \times \lambda$ patch $\mathcal{N}(p)$ centered at p , $K^{\mathcal{N}(p),c} \in \mathbb{R}^{d \times \lambda \times \lambda \times \hat{C}^k}$ and $V^{\mathcal{N}(p),c} \in \mathbb{R}^{d \times \lambda \times \lambda \times \hat{C}^v}$. Then we apply the local attention to Q_i^p with $K^{\mathcal{N}(p),c}$ and $V^{\mathcal{N}(p),c}$ and obtain the output of LA module at location p as:

$$\mathbf{L}_i^p = \text{softmax} \left(\frac{\mathbf{Q}_i^p (\mathbf{K}^{\mathcal{N}(p),c})^\top}{\beta} \right) \mathbf{V}^{\mathcal{N}(p),c}, \quad (12)$$

where $\mathbf{Q}_i^p \in \mathbb{R}^{1 \times \hat{C}^k}$, $\mathbf{K}^{\mathcal{N}(p),c} \in \mathbb{R}^{d \lambda^2 \times \hat{C}^k}$ and $\mathbf{V}^{\mathcal{N}(p),c} \in \mathbb{R}^{d \lambda^2 \times \hat{C}^v}$ are matrices obtained by reshaping Q_i^p , $K^{\mathcal{N}(p),c}$ and $V^{\mathcal{N}(p),c}$ from their original size, β is the scaling factor. Finally, we obtain the feature $L_i \in \mathbb{R}^{\hat{H} \times \hat{W} \times \hat{C}^v}$ by reshaping \mathbf{L}_i to its original size.

To restore the colors of the input frame X_i , we further adopt a simple but effective decoder. Specifically, we use a decoder $\mathcal{D}(\cdot)$ consisting of ResBlocks [6] followed by up-sampling interpolation layers to gradually refine the enhanced feature $V_i + L_i$ and obtain the predicted i -th chrominance channel Y_i as:

$$Y_i = \mathcal{D}(V_i + L_i). \quad (13)$$

4 Experimental Results

In this section, we first describe the experimental settings of the proposed ColormNet. Then we evaluate the effectiveness of our approach against state-of-the-art methods. More experimental results are included in the supplemental material. Code and models are available at <https://github.com/yyang181/colormnet>.

4.1 Experimental settings

Datasets. Following previous works [1, 20, 36], we use the datasets of DAVIS [25] and Videvo [15] for training and generate grayscale video frames using OpenCV library. For testing, we use three popular benchmark test datasets including the DAVIS validation set, the Videvo validation set and the official validation set of NTIRE 2023 Video Colorization Challenge [11] (NVCC2023 for short).

Table 1: Quantitative comparisons of the proposed method against state-of-the-art ones on the DAVIS [25] validation set (short frame length), the Videvo [15] validation set (medium frame length) and the NVCC2023 [11] validation set (long frame length). Our method achieves favorable performance in most of the metrics. Top 1_{st} and 2_{nd} results are marked in **bold red** and **blue** respectively. * denotes that we apply DVP [18] to the results of Color2Embed and DDColor as post-processing method. Note that for the LPIPS matrix on Videvo, we examine the performance of our method and BiSTNet in additional decimal places to determine the best one and the second best one as the performance of these two methods appears to be identical when displayed in the table with a limited number of decimal places. † denotes that two exemplars are used.

Methods	DDColor	Color2Embed	DDColor*	Color2Embed*	VCGAN	TCVC	DeepExemplar	DeepRemaster	BiSTNet†	ColorMNet
Categories	[12]	[43]	[12]	[43]	[44]	[20]	[39]	[9]	[36]	(Ours)
DAVIS										
Image-based										
PSNR (dB)↑	30.84	31.33	30.67	31.05	30.24	31.10	33.24	33.25	34.02	35.77
FID↓	65.13	101.08	86.96	118.11	128.48	116.41	69.56	92.28	44.69	38.39
SSIM↑	0.926	0.951	0.936	0.943	0.924	0.955	0.950	0.961	0.964	0.970
LPIPS↓	0.085	0.076	0.086	0.086	0.100	0.080	0.062	0.060	0.043	0.035
Videvo										
PSNR (dB)↑	30.76	31.65	30.60	31.62	30.62	31.29	33.11	32.95	34.12	34.35
FID↓	45.08	66.73	50.80	73.02	97.86	80.74	54.93	68.15	32.25	30.76
SSIM↑	0.925	0.958	0.940	0.960	0.934	0.956	0.956	0.964	0.968	0.972
LPIPS↓	0.079	0.060	0.077	0.061	0.085	0.068	0.053	0.053	0.036	0.036
NVCC2023										
PSNR (dB)↑	29.95	30.90	30.16	30.66	29.77	30.45	32.03	32.25	33.18	33.26
FID↓	48.50	65.92	95.83	70.35	86.59	72.76	35.39	53.03	25.55	20.16
SSIM↑	0.888	0.933	0.911	0.927	0.866	0.935	0.930	0.951	0.949	0.959
LPIPS↓	0.114	0.091	0.099	0.098	0.130	0.089	0.073	0.071	0.054	0.039

Exemplars. Following [1, 9, 36, 39], we adopt a similar strategy to utilize the first frame of each video clip as an exemplar to colorize the video clip.

Evaluation metrics. Following the experimental protocol of most existing colorization methods, we use peak signal-to-noise ratio (PSNR), structural similarity index measurement (SSIM) [34], the Fréchet Inception Distance (FID) [8], and the learned perceptual image patch similarity (LPIPS) [41] as evaluation metrics. These assessment matrices cover a spectrum of pixel-wise considerations, the distribution similarity between generated images as well as ground truth images, and the perception similarity.

Implementation details. We train our model on a machine with one RTX A6000 GPU. We adopt the Adam optimizer [14] with default parameters using PyTorch [23] for 160,000 iterations. The batch size is set to 4. We adopt the CIE LAB color space for each frame in our experiments. The learning rate is set to a constant 2×10^{-5} . We empirically set $\gamma = 5$, $N_e = 5$, $N_s = 10$, $M = 128$ and $d = 1$. We employ an L_1 loss, computed as the mean absolute errors between the predicted images and the ground truths.

4.2 Comparisons with state-of-the-art methods

Quantitative comparison. We benchmark our method against state-of-the-art ones on three datasets and report quantitative results. The competing methods include the automatic colorization techniques [20, 44], single exemplar-based approaches [9, 39], and a double exemplar-based method [36]. Furthermore, for enhanced self-containment, we incorporate comparisons with leading image colorization methods [12, 43] and enhance the temporal consistency of these methods by applying DVP [18] as post-processing method. For [9, 20, 36, 44], whose weights

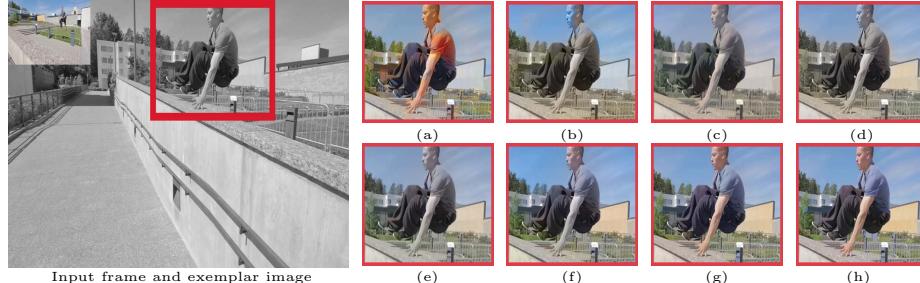


Fig. 3: Qualitative comparisons on clip *parkour* from the validation set of DAVIS [25] dataset. (a)-(g) are the colorization results by DDColor [12], TCVC [20], VCGAN [44], DeepRemaster [9], DeepExemplar [39], BiSTNet[†] [36] and ColorMNet (Ours). (h) Ground truth. The evaluated methods do not generate realistic colorful images in (a)-(f). In contrast, our approach generates a well-colorized image in (g).

are trained on the same datasets as our method, we conduct tests using their official codes and weights provided by the authors. However, for [12, 39, 43], we train from scratch using the official codes on identical datasets as our method to ensure fair comparisons. In addition, we adhere to the commonly adopted protocol of using the same first frame ground truth image of each video clip as the exemplar for testing all exemplar-based methods [9, 36, 39, 43], with the exception that we also include the last frame as an extra exemplar for testing the double exemplar-based method, BiSTNet [36].

Table 1 shows that the ColorMNet consistently generates competitive colorization results on the DAVIS [25] validation set, the Videvo [15] validation set, and the NVCC2023 [11] validation set, where our method performs better than the evaluated methods in terms of PSNR, SSIM, FID, and LPIPS, indicating that our method not only can generate both high-quality and high-fidelity colorization results, but also demonstrates the good generalization based on the favorable performance on the validation set in NVCC2023 (the training set in NVCC2023 is not included in the training phase).

Qualitative evaluation. Figure 3(a) shows that the image-based method [12] generates results with non-uniform colors on the man’s face and cloth. Automatic video colorization techniques [20, 44] can not generate vivid colors (Figure 3(b) and (c)). Exemplar-based methods [9, 36, 39] can not establish long-range correspondence and thus fail to restore the colors on the man’s arms and cloth (Figure 3(d)-(f)). In contrast, the proposed ColorMNet generates a vivid colorized image (Figure 3(g)) by modeling both the spatial information from each frame and the temporal information from far-apart frames.

Evaluations on real-world videos. We further evaluate the proposed method on a real-world grayscale video, *Manhattan (1979)*. We obtain the exemplars (Figure 4(b)) by searching the internet to find the most visually similar images to the input video frames. Figure 4(c) and (d) show that state-of-the-art methods [9, 39] do not colorize the objects (*e.g.*, the wall of the building, the trees and the sky) well. In contrast, our method generates better-colorized frames, where the colors look natural and realistic (Figure 4(e)). In addition, our method

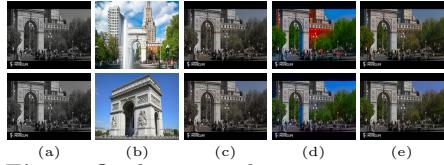


Fig. 4: Qualitative colorization comparisons on real-world video *Manhattan* (1979). (a) Input frame. (b) Exemplar images obtained by Google Image Search. (c)-(e) are the colorization results by DeepRemaster [9], DeepExemplar [39] and ColorMNet (Ours), respectively. The methods [9, 39] do not colorize the wall of the building, the trees, and the sky well in (c) and (d). Our ColorMNet generates error-free and realistic colors in (e).

demonstrates its robustness by consistently generating similar results even when provided with exemplar images that possess diverse colors and contents.

Efficiency evaluation. Given that practical applications of video colorization often involve processing longer videos, where maximum GPU memory usage and inference speed are critical metrics, we further evaluate our method against three representative state-of-the-art exemplar-based video colorization approaches [9, 36, 39]. Specifically, we record the maximum GPU memory consumption during the inference on a machine with an NVIDIA RTX A6000 GPU. The average running time is obtained using 300 test images with a 960×536 resolution.

Table 2 shows that the maximum GPU consumption of ColorMNet (ours) is only 11.2% of DeepRemaster [9], 10.0% of DeepExemplar [39] and 5.4% of BiSTNet [36]; the running time is at least $8\times$ faster than the evaluated methods.

Temporal consistency evaluations. To examine whether the colorized videos generated by our method have a better temporal consistency property, we use the color distribution consistency index (CDC) [20] as the metric. Table 2 shows that our method has a lower CDC value when compared to exemplar-based methods [9, 36, 39] on the DAVIS [25] validation set, which indicates that our method is capable of generating videos with improved temporal consistency by exploring better temporal information.

5 Analysis and Discussion

To better understand how our method solves video colorization and demonstrate the effectiveness of its main components, we conduct a deeper analysis of the proposed approach. For the ablation studies in this section, we train our method and all alternative baselines on the training set of the DAVIS [25] dataset and the Videvo [15] dataset with 160,000 iterations for fair comparisons.

Effectiveness of PVGFE. The proposed PVGFE explores robust spatial features that can model both global semantic structures and local details for better video colorization. To demonstrate its effectiveness, we compare with baseline methods that respectively replace the PVGFE with the pretrained ResNet50 [6]

Table 2: Quantitative evaluations of the video colorization methods with better accuracy performance on the DAVIS [25] dataset in terms of maximum GPU memory consumption, average running time and temporal consistency index CDC.

Methods	Memory (G)	Running time (/s)	CDC \downarrow
DeepExemplar [39]	19.0	0.80	0.003876
DeepRemaster [9]	16.6	0.61	0.004285
BiSTNet [36]	34.9	1.62	0.003870
ColorMNet (Ours)	1.9	0.07	0.003763

Table 3: Effectiveness of the proposed PVGFE, MFP, and LA modules in our ColorMNet. Evaluated on the DAVIS [25].

Components	Feature extractor	Feature propagation	Locality	Metrics
Methods	ResNet50/DINOv2/PVGFE	Stacking Recurrent MFP	LA	PSNR/SSIM
ColorMNet _{w/o} /ResNet50	✓		✓	35.77/0.962
ColorMNet _{w/o} /DINOv2	✓	✓	✓	35.38/0.963
ColorMNet _{w/o} /colorization	✓	✓	✓	35.26/0.965
ColorMNet _{w/o} /Stacking	✓	✓	✓	33.94/0.961
ColorMNet _{w/o} /Recurrent	✓	✓	✓	35.26/0.966
ColorMNet _{w/o} /LA	✓	✓	✓	35.44/0.967
ColorMNet (Ours)	✓	✓	✓	35.77/0.970

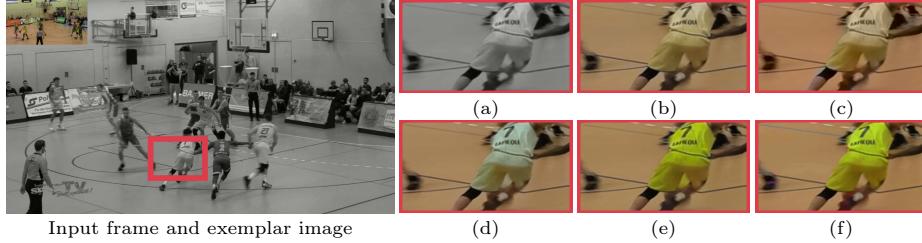


Fig. 5: Effectiveness of PVGFE for video colorization. (a) Input patch. (b)-(e) are the colorization results by ColorMNet_{w/ ResNet50}, ColorMNet_{w/ DINov2}, ColorMNet_{w/ Concatenation} and ColorMNet (Ours), respectively. (f) Ground truth. Compared to the baselines, our approach yields a more natural colorized result in (e).

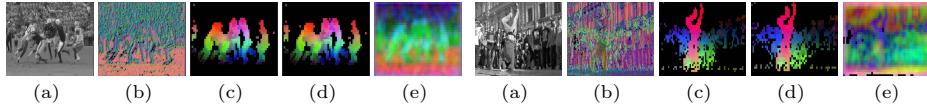


Fig. 6: Visualization of features. We use the PCA tools by [22]. (a) Input frame. (b)-(e) are the features generated by the feature extractors of ColorMNet_{w/ ResNet50}, ColorMNet_{w/ DINov2}, ColorMNet_{w/ Concatenation} and ColorMNet (Ours), respectively. Compared with (b), (c) and (d), our proposed PVGFE can generate features that are not only semantic-aware (*i.e.*, the players and the dancer in the foreground) but also sensitive to local details (*i.e.*, a crowd of spectators in the background) in (e).

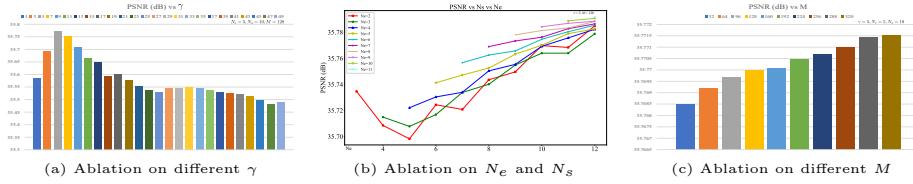


Fig. 7: Extensive ablation study on the detailed design of the proposed MFP module.

(ColorMNet_{w/ ResNet50} for short), the pretrained DINov2 [22] (ColorMNet_{w/ DINov2} for short), and the concatenation of both pretrained ResNet50 and DINov2 (ColorMNet_{w/ Concatenation} for short) in our implementation. Table 3 shows that our ColorMNet with the PVGFE outperforms all baseline methods. The qualitative comparisons in Figure 5 show that the results obtained by baseline methods exhibit severe color distortions on the cloth of the player (Figure 5(b)-(d)). In contrast, our proposed ColorMNet with the PVGFE generates a better-colorized frame in Figure 5(e). Note that the PVGFE can adaptively enhance useful features while reducing the influence of useless information based on the similarities computed on input features by employing cross-attention (*i.e.*, (2)). However, a direct concatenation of features evenly without discrimination, *i.e.*, ColorMNet_{w/ Concatenation}, is less effective for reducing the impact of useless features, thus degrading performance in Table 3 and Figure 5(d).

To better understand the feature estimators mentioned above, we use the PCA tools by [22] to visualize the features generated by them. Figure 6(b) shows that ResNet50 cannot generate features that are aware of semantic structures. Although DINov2 generates more semantic features in Figure 6(c) and (d), it lacks the local details vital for colorization tasks, which explains why DINov2

performs favorably in high-level vision tasks, *i.e.*, classification and segmentation (see [22] for details), but fails in colorization as the exact colors for pixels of objects are crucial considerations in colorization, unlike in segmentation where the primary decision is whether or not a pixel belongs to a human. Figure 6(e) shows that our proposed PVGFE module is capable of generating better features optimized for colorization, retaining both semantic relevance and local details.

Effectiveness of MFP. The proposed MFP propagates temporal features for better long-range correspondences. To investigate whether directly stacking multiple frames along the temporal dimension or recurrently propagating features can already generate competitive results, we compare with baseline methods that respectively replace the MFP with direct stacking of the features from all previous colorized frames and the exemplar image along the temporal dimension (ColorMNet_{w/ Stacking} for short) and the recurrent-based feature propagation [17, 32, 33, 39] (ColorMNet_{w/ Recurrent} for short) in our implementation.

Table 3 shows that the PSNR value of our ColorMNet is at least 0.51dB higher than each baseline method, which illustrates the effectiveness of the proposed MFP in propagating features for video colorization. Figure 8(b) shows that the baseline that directly stacks frames is not able to generate a realistic image as spatial-temporal priors are not well-explored. The result obtained by the baseline with recurrent-based feature propagation contains significant color distortions on the boy (Figure 8(c)), as the errors accumulate in the recurrent-based propagation steps. In contrast, the proposed ColorMNet using the MFP generates a vivid and error-free image in Figure 8(d).

We further conduct an extensive ablation study on the parameters of the proposed MFP module. Figure 7(a) shows that our method achieves its peak PSNR when γ equals 5, as a higher γ risks potential information loss, while a lower γ could contribute redundant data. Figure 7(b) and (c) show that our method generally achieves slightly higher PSNR values with N_e , N_s and M increasing, respectively. However, note that the GPU memory usage escalates correspondingly with larger values of N_e , N_s and M .

Efficiency of MFP. To examine the efficiency of the proposed MFP, we further evaluate the proposed ColorMNet against the baseline method with direct stacking (*i.e.*, ColorMNet_{w/ Stacking}) on the validation set of NVCC2023 [11] in terms of the maximum GPU memory consumption and the average running time. Table 4 shows that our ColorMNet only requires 7.8% of the maximum GPU consumption of the baseline method, but the average running time of our approach is nearly 14× faster than the baseline method, which indicates the efficiency of the proposed MFP.

Tables 3 and 4 show that our approach using the MFP achieves a favorable performance in terms of faster inference speed, lower GPU memory consumption, and better colorization results, which demonstrates the effectiveness and efficiency of the proposed MFP in video colorization.

Effectiveness of LA. To demonstrate the effect of the proposed LA, we further compare with a baseline method that removes the LA module (ColorMNet_{w/o LA} for short) in our implementation. Table 3 shows that our ColorMNet using the

Table 4: Efficiency of the proposed MFP module for video colorization.

Methods	Memory consumption (G)	Running time (/s)
ColorMNet _{w/ Stacking}	24.1	1.04
ColorMNet (Ours)	1.9	0.07



Fig. 8: Effectiveness of the MFP module for video colorization. (a) Input frame and exemplar image. (b)-(d) are the colorization results by ColorMNet_{w/ Stacking}, ColorMNet_{w/ Recurrent} and ColorMNet (Ours), respectively. (e) Ground truth. Compared with (b) and (c), the colorized result (d) by our method contains fewer color distortions and more vivid details.

LA generates better results with higher PSNR and SSIM values than the baseline method. Figure 9(b) shows that the baseline method without the LA does not exploit the prior information among consecutive frames and thus cannot restore the colors on the sky and the leaves. However, our approach generates vivid and realistic colors in Figure 9(c), which demonstrates that the proposed LA module is effective in capturing and leveraging better spatial-temporal features.

Closely-related methods. To the best of our knowledge, we are the first to optimize a memory bank strategy suitable for colorization, yet it should be acknowledged that related strategies have been explored in some video processing works, *e.g.*, MAMBA [30] constructs a memory bank to solve video object detection by employing random selection strategy, MeMOTR [5] introduces a long-term memory to solve video object tracking by assigning exponentially decaying weights to it. Unlike MAMBA which applies a randomized selection approach, treating every feature on par, or MeMOTR which updates past memorized features via exponentially decaying weights, our proposed MFP module stores features based on their importance which is determined by the frequency of usage, thus empowering the ability of global relation mining.

We further adopt the random selection in MAMBA and the decaying weights in MeMOTR to replace our MFP for comparison. To ensure a fair comparison, the same training settings are kept for model testing. Figure 10 shows that our method can generate better colors for the dancing girl and the green grass.

Limitations. Our proposed method aims to further enhance video colorization performance while reducing GPU memory usage. However, there are limitations in the model complexity, *i.e.*, our model requires 123.61 Million parameters.

6 Conclusion

We present an effective memory-based deep spatial-temporal feature propagation network for video colorization. We develop a large-pretrained visual model guided feature estimation module to better explore robust spatial features. To establish reliable connections from far-apart frames, we propose a memory-based



Fig. 9: Effectiveness of the proposed LA module for video colorization. (a) Input frame and exemplar image. (b) and (c) are the colorization results by ColorMNet_{w/o LA} and ColorMNet (Ours). (d) Ground truth. Our approach is able to generate better-colored result in (c).



Fig. 10: Comparison results with closely-related methods on the DAVIS [25].

feature propagation module. We develop a local attention module to better utilize spatial-temporal priors. Both quantitative and qualitative experimental results show that our method performs favorably against state-of-the-art methods.

References

1. Chen, S., Li, X., Zhang, X., Wang, M., Zhang, Y., Han, J., Zhang, Y.: Exemplar-based video colorization with long-term spatiotemporal dependency. arXiv preprint arXiv:2303.15081 (2023)
2. Chen, X., Zou, D., Zhao, Q., Tan, P.: Manifold preserving edit propagation. ACM TOG **31**(6), 1–7 (2012)
3. Cheng, Z., Yang, Q., Sheng, B.: Deep colorization. In: ICCV (2015)
4. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. In: ICLR (2021)
5. Gao, R., Wang, L.: MeMOTR: Long-term memory-augmented transformer for multi-object tracking. In: ICCV (2023)
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
7. He, M., Chen, D., Liao, J., Sander, P.V., Yuan, L.: Deep exemplar-based colorization. ACM TOG **37**(4), 1–16 (2018)
8. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. In: NeurIPS (2017)
9. Iizuka, S., Simo-Serra, E.: Deepremaster: temporal source-reference attention networks for comprehensive video enhancement. ACM TOG **38**(6), 1–13 (2019)
10. Iizuka, S., Simo-Serra, E., Ishikawa, H.: Let there be color! joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification. ACM TOG **35**(4), 1–11 (2016)
11. Kang, X., Lin, X., Zhang, K., et al.: Ntire 2023 video colorization challenge. In: CVPRW (2023)
12. Kang, X., Yang, T., Ouyang, W., Ren, P., Li, L., Xie, X.: Ddcolor: Towards photo-realistic image colorization via dual decoders. In: ICCV (2023)
13. Karen Simonyan, A.Z.: Very deep convolutional networks for large-scale image recognition. In: ICLR (2015)
14. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: ICLR (2015)
15. Lai, W.S., Huang, J.B., Wang, O., Shechtman, E., Yumer, E., Yang, M.H.: Learning blind video temporal consistency. In: ECCV (2018)
16. Larsson, G., Maire, M., Shakhnarovich, G.: Learning representations for automatic colorization. In: ECCV (2016)
17. Lei, C., Chen, Q.: Fully automatic video colorization with self-regularization and diversity. In: CVPR (2019)
18. Lei, C., Xing, Y., Chen, Q.: Blind video temporal consistency via deep video prior. In: NeurIPS (2020)
19. Levin, A., Lischinski, D., Weiss, Y.: Colorization using optimization. ACM TOG **23**(3), 689–694 (2004)
20. Liu, Y., Zhao, H., Chan, K.C., Wang, X., Loy, C.C., Qiao, Y., Dong, C.: Temporally consistent video colorization with deep feature propagation and self-regularization learning. arXiv preprint arXiv:2110.04562 (2021)

21. Luan, Q., Wen, F., Cohen-Or, D., Liang, L., Xu, Y., Shum, H.: Natural image colorization. In: ESRT (2007)
22. Oquab, M., Darzet, T., Moutakanni, T., et al.: Dinov2: Learning robust visual features without supervision. TMLR (2024)
23. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. In: NeurIPS (2019)
24. Paul, S., Bhattacharya, S., Gupta, S.: Spatiotemporal colorization of video using 3d steerable pyramids. IEEE TCSVT **27**(8), 1605–1619 (2016)
25. Perazzi, F., Pont-Tuset, J., McWilliams, B., Van Gool, L., Gross, M., Sorkine-Hornung, A.: A benchmark dataset and evaluation methodology for video object segmentation. In: CVPR (2016)
26. Qu, Y., Wong, T., Heng, P.: Manga colorization. ACM TOG **25**(3), 1214–1220 (2006)
27. Sangkloy, P., Lu, J., Fang, C., Yu, F., Hays, J.: Scribbler: Controlling deep image synthesis with sketch and color. In: CVPR (2017)
28. Sheng, B., Sun, H., Magnor, M., Li, P.: Video colorization using parallel optimization in feature space. IEEE TCSVT **24**(3), 407–417 (2013)
29. Su, J.W., Chu, H.K., Huang, J.B.: Instance-aware image colorization. In: CVPR (2020)
30. Sun, G., Hua, Y., Hu, G., Robertson, N.: Mamba: Multi-level aggregation via memory bank for video object detection. In: AAAI (2021)
31. Teed, Z., Deng, J.: Raft: Recurrent all-pairs field transforms for optical flow. In: ECCV (2020)
32. Thasarathan, H., Nazeri, K., Ebrahimi, M.: Automatic temporally coherent video colorization. In: CRV (2019)
33. Wan, Z., Zhang, B., Chen, D., Liao, J.: Bringing old films back to life. In: CVPR (2022)
34. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE TIP **13**(4), 600–612 (2004)
35. Xu, Z., Wang, T., Fang, F., Sheng, Y., Zhang, G.: Stylization-based architecture for fast deep exemplar colorization. In: CVPR (2020)
36. Yang, Y., Peng, Z., Du, X., Tao, Z., Tang, J., Pan, J.: Bistnet: Semantic image prior guided bidirectional temporal feature fusion for deep exemplar-based video colorization. IEEE TPAMI pp. 1–14 (2024)
37. Yatziv, L., Sapiro, G.: Fast image and video colorization using chrominance blending. IEEE TIP **15**(5), 1120–1129 (2006)
38. Zamir, S.W., Arora, A., Khan, S., Hayat, M., Khan, F.S., Yang, M.H.: Restormer: Efficient transformer for high-resolution image restoration. In: CVPR (2022)
39. Zhang, B., He, M., Liao, J., Sander, P.V., Yuan, L., Bermak, A., Chen, D.: Deep exemplar-based video colorization. In: CVPR (2019)
40. Zhang, R., Isola, P., Efros, A.A.: Colorful image colorization. In: ECCV (2016)
41. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR (2018)
42. Zhang, R., Zhu, J.Y., Isola, P., Geng, X., Lin, A.S., Yu, T., Efros, A.A.: Real-time user-guided image colorization with learned deep priors. ACM TOG **36**(4), 1–11 (2017)
43. Zhao, H., Wu, W., Liu, Y., He, D.: Color2embed: Fast exemplar-based image colorization using color embeddings. arXiv preprint arXiv:2106.08017 (2021)

44. Zhao, Y., Po, L.M., Yu, W.Y., Rehman, Y.A.U., Liu, M., Zhang, Y., Ou, W.: Vgan: video colorization with hybrid generative adversarial network. IEEE TMM (2022)

ColorMNet: A Memory-based Deep Spatial-Temporal Feature Propagation Network for Video Colorization

Yixin Yang, Jiangxin Dong, Jinhui Tang, and Jinshan Pan

Nanjing University of Science and Technology

Overview

In this document, we first present the network details in Section 1. Then, we analyze the effectiveness of the proposed large-pretrained visual model guided feature estimation (PVGFE) module and the memory-based feature propagation (MFP) module in Section 2 and Section 3. To examine the effectiveness of the proposed local attention (LA) module on video colorization, we further analyze it in Section 4. In addition, we conduct a user study to investigate the subjective preference by human observers of each colorization method in Section 5. Finally, we show more visual comparisons on both synthetic datasets and real-world videos in Section 6.

1 Network Details

As stated in Section 3 of the main manuscript, our method contains a large-pretrained visual model guided feature estimation module, a memory-based feature propagation module, and a local attention module for video colorization. We also show the network details of the proposed memory-based deep spatial-temporal feature propagation network for video colorization in Figures 2 of the main manuscript. In this document, we list the detailed architecture of our proposed ColorMNet in Table 1. The spatial resolution of the input image is 448×448 pixels.

2 Effectiveness of the Large-Pretrained Visual Model Guided Feature Estimation Module

As stated in Section 5 of the manuscript, we have analyzed the effectiveness of the large-pretrained visual model guided feature estimation (PVGFE) module. In this supplemental material, we further show more visual comparisons to demonstrate the effectiveness of the PVGFE module. In ‘*Comparisons With-SOTA.mp4*’, we show that our proposed ColorMNet with using the PVGFE is able to generate better-colorized videos.

Table 1: Detailed architecture of our proposed ColorMNet. [Conv. 7×7 , 64, stride 2] denotes a convolution with the filter size of 7×7 pixels with the filter number of 64 with stride 2, Embed dim. denotes the dimension of embedding, [Interpolation, $\times 2$] denotes an interpolation operation with a scale factor equal to 2, [ResBlock, 256] denotes a ResBlock consisting of convolutions with the filter size of 3×3 pixels with the filter number of 256.

		ColorMNet	
		Output size	ResNet50 [2] DINov2 [8]
Key feature extractor (PVGFE)	28×28×1024	Conv. 7×7 , 64, stride 2	Patch Embedding
		MaxPool, 3×3 , stride 2	[Transformer block Patch size = 14 Embed dim. = 384] $\times 12$
		$\begin{bmatrix} \text{Conv. } 1 \times 1, 64 \\ \text{Conv. } 3 \times 3, 64 \\ \text{Conv. } 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} \text{Heads} = 6 \\ \text{Blocks} = 12 \end{bmatrix}$
		$\begin{bmatrix} \text{Conv. } 1 \times 1, 128 \\ \text{Conv. } 3 \times 3, 128 \end{bmatrix} \times 4$	FFN layer = MLP
		$\begin{bmatrix} \text{Conv. } 1 \times 1, 512 \\ \text{Conv. } 1 \times 1, 256 \\ \text{Conv. } 3 \times 3, 256 \end{bmatrix} \times 6$	Concat features from last 4 layers
		$\begin{bmatrix} \text{Conv. } 1 \times 1, 1536 \\ \text{Conv. } 3 \times 3, 1024 \end{bmatrix}$	Conv. 1×1 , 1536
		Conv. 3×3 , 64	Interpolation, $\times 14/16$
		Conv. 3×3 , 64	Conv. 3×3 , 1024
		Cross-channel attention [11]	
			ResNet18 [2]
Value feature extractor (ResNet18 [2])	28×28×256	Conv. 7×7 , 64, stride 2	
		MaxPool, 3×3 , stride 2	
		$\begin{bmatrix} \text{Conv. } 1 \times 1, 64 \\ \text{Conv. } 3 \times 3, 64 \end{bmatrix} \times 2$	
		$\begin{bmatrix} \text{Conv. } 1 \times 1, 128 \\ \text{Conv. } 3 \times 3, 128 \end{bmatrix} \times 2$	
		$\begin{bmatrix} \text{Conv. } 1 \times 1, 256 \\ \text{Conv. } 3 \times 3, 256 \end{bmatrix} \times 2$	
Decoder	112×112×256	Conv. 3×3 , 512	
		Interpolation, $\times 2$	
		ResBlock, 256	
		Interpolation, $\times 2$	
		ResBlock, 256	
	112×112×2	Conv. 1×1 , 2	
	112×112×2	GRU [1]	
	448×448×2	Interpolation, $\times 4$	

3 Effectiveness of the Memory-based Feature Propagation Module

As stated in Section 5 of the manuscript, we have analyzed the effectiveness of the memory-based feature propagation (MFP) module. In this supplemental material, we further show more visual comparisons to demonstrate the effectiveness of the MFP module. In ‘ComparisonsWithSOTA.mp4’, we show that our proposed ColorMNet with using the MFP is able to generate better-colorized videos compared with the method without using the MFP.

4 Effectiveness of the Local Attention Module

As stated in Section 5 of the manuscript, we have analyzed the effectiveness of the local attention (LA) module. We empirically set $\lambda = 7$ for $\lambda \times \lambda$ patch $\mathcal{N}(p)$. In

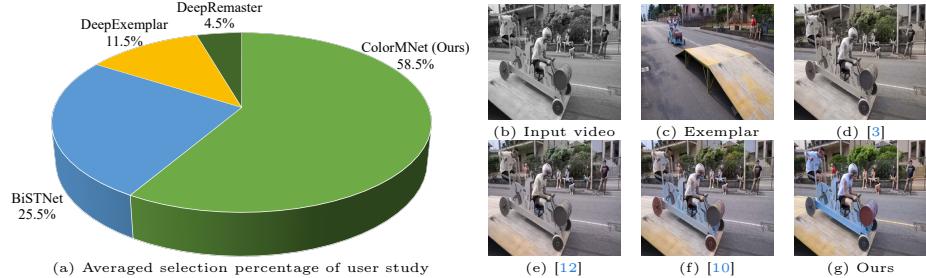


Fig. 1: User study result and an example of a group of results displayed to human observers in the user study. (a) shows that our proposed ColorMNet achieves obviously higher score than other state-of-the-art methods, which demonstrates its subjective advantages. (b)-(g) are the input video, the exemplar image, the colorized videos by DeepRemaster [3], DeepExemplar [12], BiSTNet[†] [10] and ColorMNet (Ours), respectively. We make the methods anonymous and randomly sort the videos in (d)-(g) to ensure fairness. [†] denotes that two exemplars are used.

this supplemental material, we further show more visual comparisons to demonstrate the effectiveness of the LA module. In ‘ComparisonsWithSOTA.mp4’, we show that our proposed ColorMNet with using the LA is able to generate better-colorized videos.

5 User Study

To evaluate whether our results are favored by human observers, we further conduct user study experiments. Specifically, we compare our method with exemplar-based methods, i.e., BiSTNet [10], DeepExemplar [12] and DeepRemaster [3]. We randomly select 10 input videos from the DAVIS [9] validation set, the Videvo [6] validation set and the NVCC2023 [4] validation set together with the colorization results and the exemplar images displayed to 20 online observers without constraints. We make the methods anonymous and randomly sort the videos in each group to ensure fairness. Observers are asked to choose the most visually pleasing results from a group of videos. Figure 1 shows that our method is preferred by a wider range of users than other state-of-the-art methods.

6 More Experimental Results

In this section, we provide more visual comparisons with state-of-the-art methods on both synthetic and real-world videos. Figures 2-13 show the comparisons, where our method generates better colorized frames. In ‘ComparisonsWithSOTA.mp4’, we show that the proposed method generates vivid and realistic videos.

References

1. Cho, K., Van Merriënboer, B., Bahdanau, D., Bengio, Y.: On the properties of neural machine translation: Encoder-decoder approaches. arXiv preprint arXiv:1409.1259 (2014)
2. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
3. Iizuka, S., Simo-Serra, E.: Deepremaster: temporal source-reference attention networks for comprehensive video enhancement. ACM TOG **38**(6), 1–13 (2019)
4. Kang, X., Lin, X., Zhang, K., et al.: Ntire 2023 video colorization challenge. In: CVPRW (2023)
5. Kang, X., Yang, T., Ouyang, W., Ren, P., Li, L., Xie, X.: Ddcolor: Towards photo-realistic image colorization via dual decoders. In: ICCV (2023)
6. Lai, W.S., Huang, J.B., Wang, O., Shechtman, E., Yumer, E., Yang, M.H.: Learning blind video temporal consistency. In: ECCV (2018)
7. Liu, Y., Zhao, H., Chan, K.C., Wang, X., Loy, C.C., Qiao, Y., Dong, C.: Temporally consistent video colorization with deep feature propagation and self-regularization learning. arXiv preprint arXiv:2110.04562 (2021)
8. Oquab, M., Dariseti, T., Moutakanni, T., et al.: Dinov2: Learning robust visual features without supervision. TMLR (2024)
9. Perazzi, F., Pont-Tuset, J., McWilliams, B., Van Gool, L., Gross, M., Sorkine-Hornung, A.: A benchmark dataset and evaluation methodology for video object segmentation. In: CVPR (2016)
10. Yang, Y., Peng, Z., Du, X., Tao, Z., Tang, J., Pan, J.: Bistnet: Semantic image prior guided bidirectional temporal feature fusion for deep exemplar-based video colorization. IEEE TPAMI pp. 1–14 (2024)
11. Zamir, S.W., Arora, A., Khan, S., Hayat, M., Khan, F.S., Yang, M.H.: Restormer: Efficient transformer for high-resolution image restoration. In: CVPR (2022)
12. Zhang, B., He, M., Liao, J., Sander, P.V., Yuan, L., Bermak, A., Chen, D.: Deep exemplar-based video colorization. In: CVPR (2019)
13. Zhao, H., Wu, W., Liu, Y., He, D.: Color2embed: Fast exemplar-based image colorization using color embeddings. arXiv preprint arXiv:2106.08017 (2021)
14. Zhao, Y., Po, L.M., Yu, W.Y., Rehman, Y.A.U., Liu, M., Zhang, Y., Ou, W.: Vcgan: video colorization with hybrid generative adversarial network. IEEE TMM (2022)



Fig. 2: Colorization results on clip *bike-packing* from the DAVIS validation dataset [9]. The results shown in (b) and (c) still contain significant color-bleeding artifacts. [3, 7, 12, 14] do not recover the man well. In contrast, our proposed method generates a better-colorized frame, where the man is restored well and the colors look better.

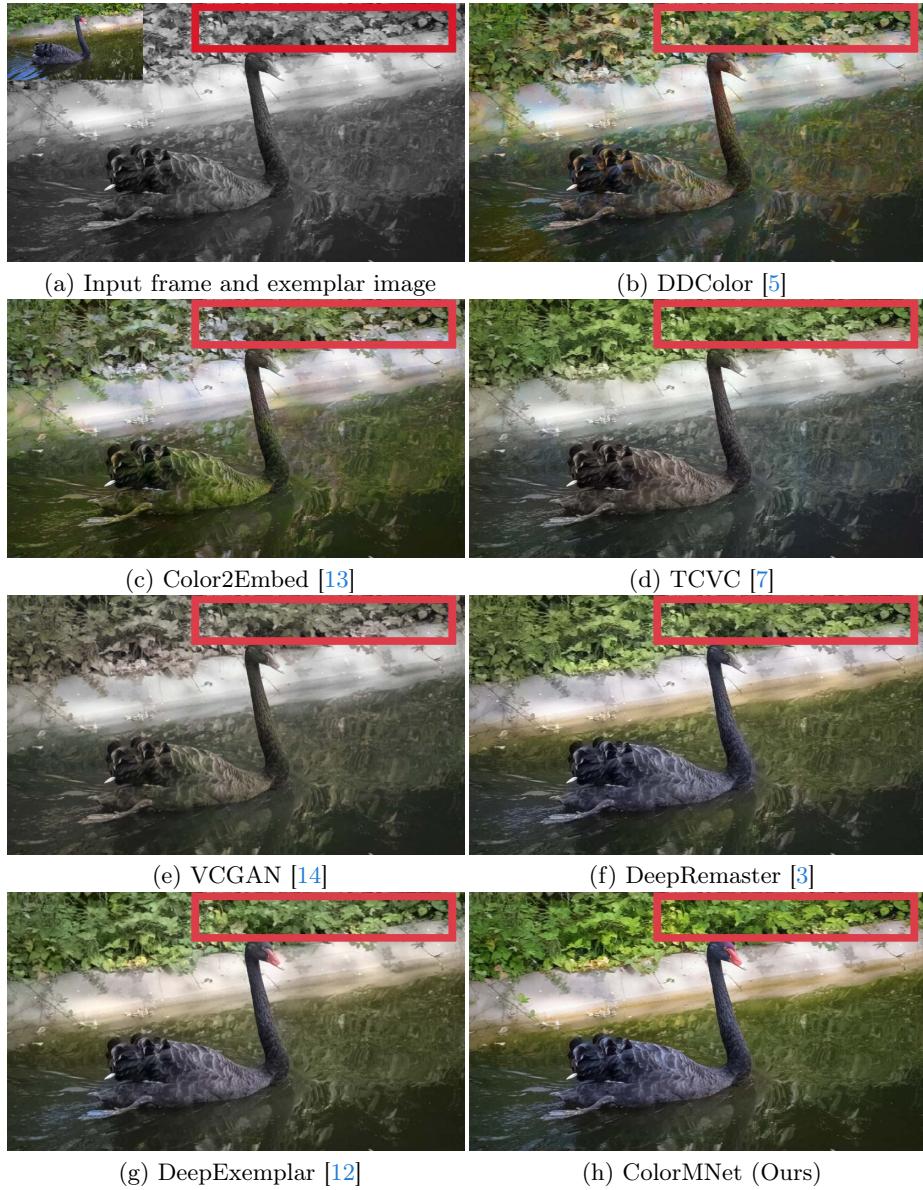


Fig. 3: Colorization results on clip *blackswan* from the DAVIS validation dataset [9]. The results shown in (b) and (c) still contain significant color-bleeding artifacts. In contrast, our proposed method generates a better-colorized frame.



Fig. 4: Colorization results on clip *breakdance* from the DAVIS validation dataset [9]. The results shown in (b) and (c) still contain significant color-bleeding artifacts. In contrast, our proposed method generates a better-colorized frame, where the colors of the dancer are restored well.

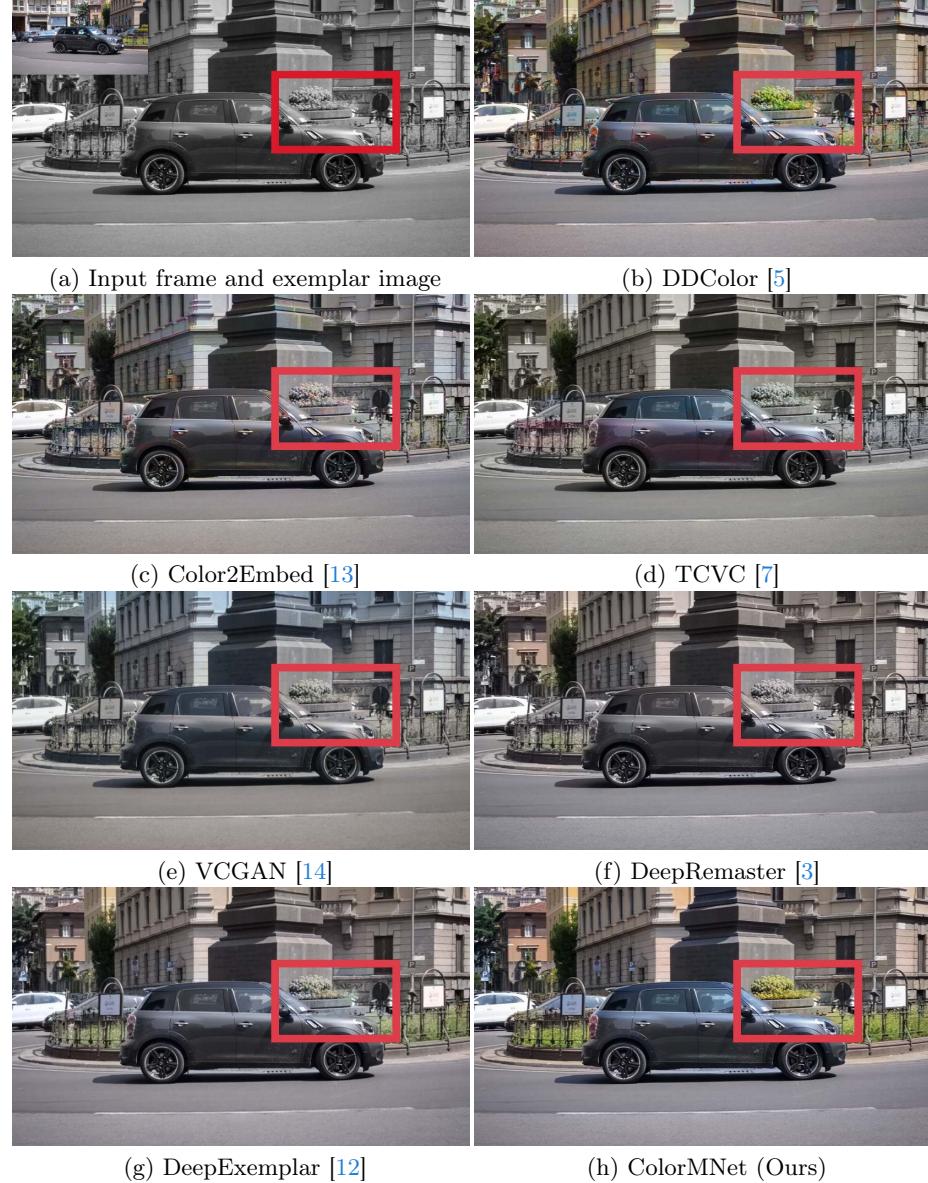


Fig. 5: Colorization results on clip *car-roundabout* from the DAVIS validation dataset [9]. The results shown in (b) and (c) still contain significant color-bleeding artifacts. In contrast, our proposed method restores the colors of the flowerbed and generates a better-colorized frame.



Fig. 6: Colorization results on clip *loading* from the DAVIS validation dataset [9]. The results shown in (b) and (c) still contain significant color-bleeding artifacts. In contrast, our proposed method generates a vivid and realistic frame, where the colors of the box and the man’s hands are better restored.



Fig. 7: Colorization results on clip *CoupleRidingMotorbike* from the Videvo validation dataset [6]. The results shown in (b) and (c) still contain significant color-bleeding artifacts. In contrast, our proposed method generates a realistic frame that is faithful to the exemplar image.



Fig. 8: Colorization results on clip *Cycling* from the Videvo validation dataset [6]. The results shown in (b) and (c) still contain significant color-bleeding artifacts. In contrast, our proposed method generates a vivid frame than other stage-of-the-art methods.

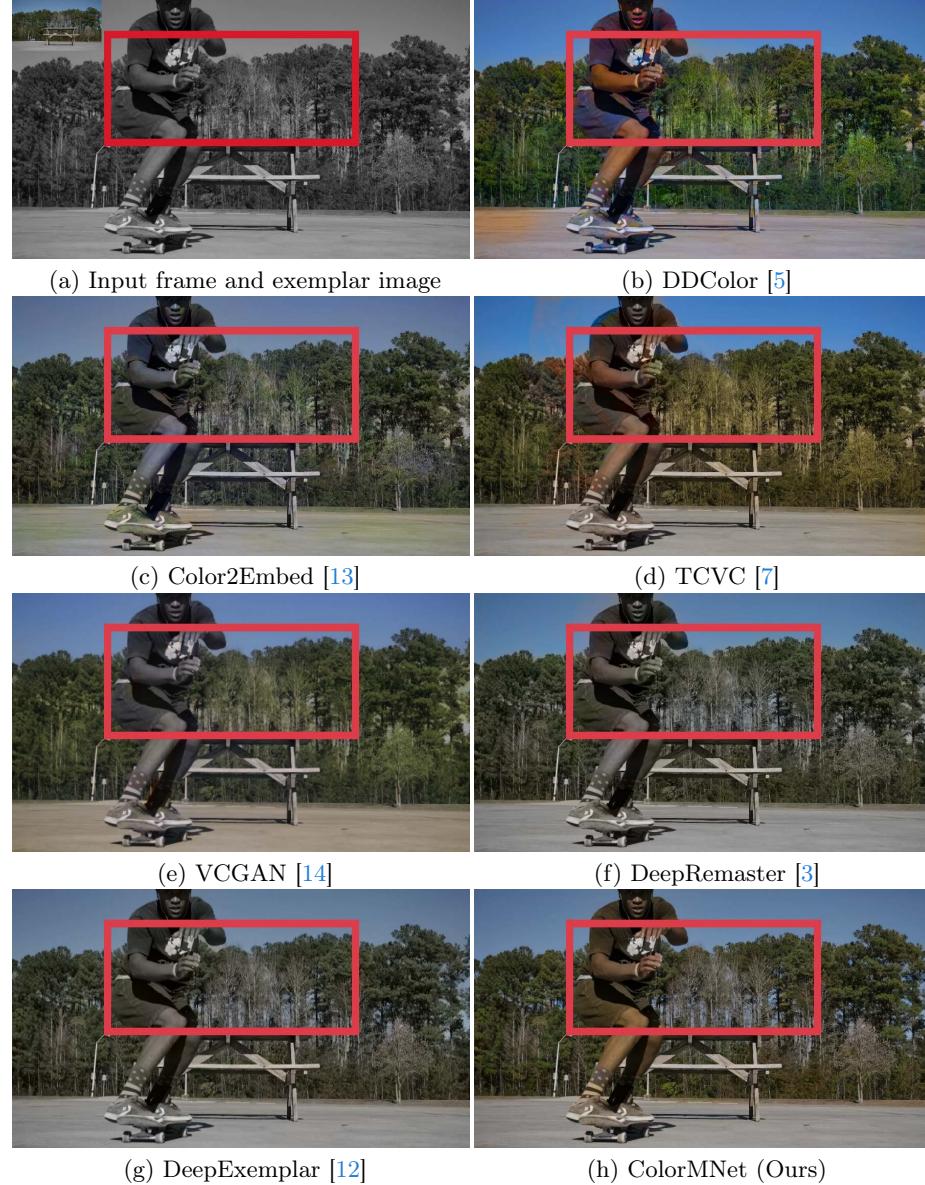


Fig. 9: Colorization results on clip *SkateboarderTableJump* from the Videvo validation dataset [6]. The results shown in (b) and (c) still contain significant color-bleeding artifacts. In contrast, our proposed method generates a realistic frame, where the colors of the skateboard man and the trees are better restored.

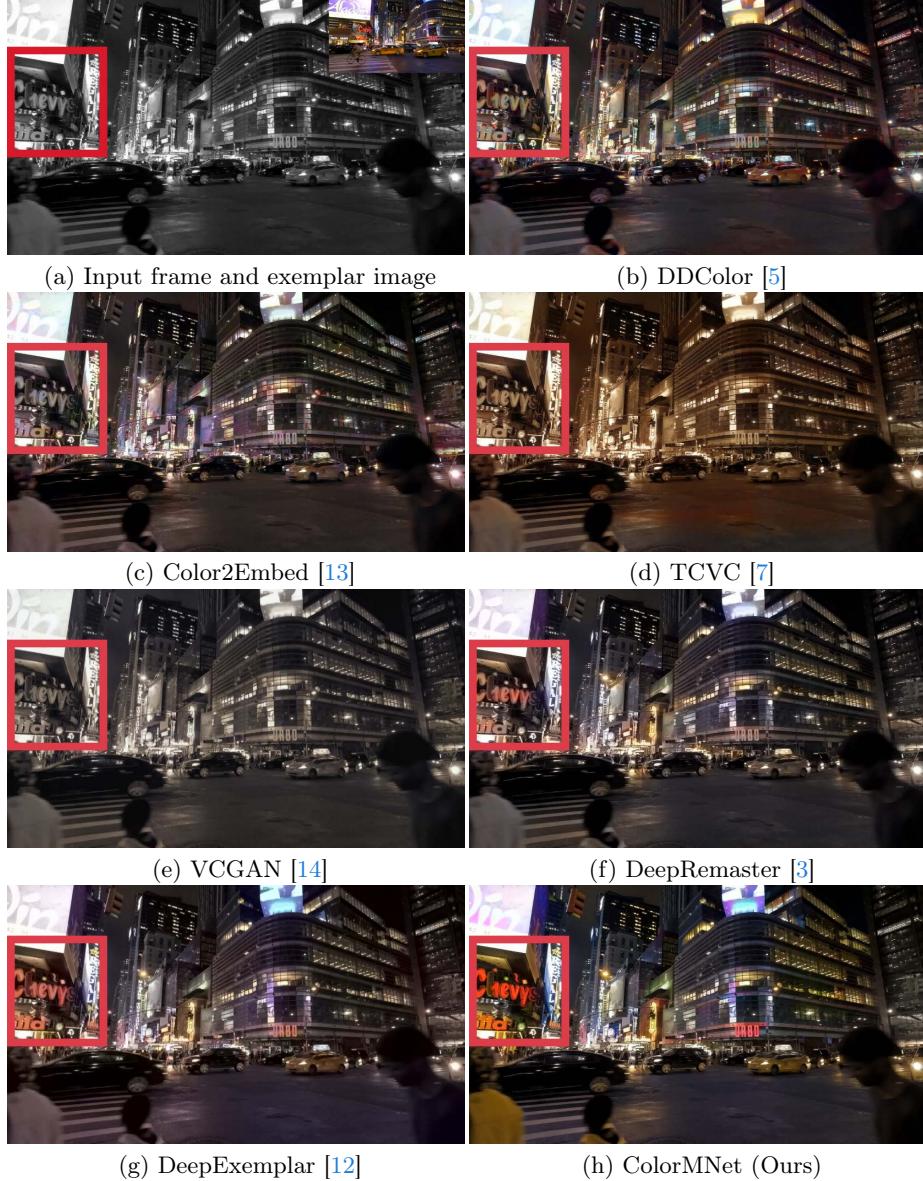


Fig. 10: Colorization results on clip *TimeSquareTraffic* from the Videvo validation dataset [6]. The results shown in (b) and (c) still contain significant color-bleeding artifacts. In contrast, our proposed method generates a vivid and realistic frame against other stage-of-the-art methods.

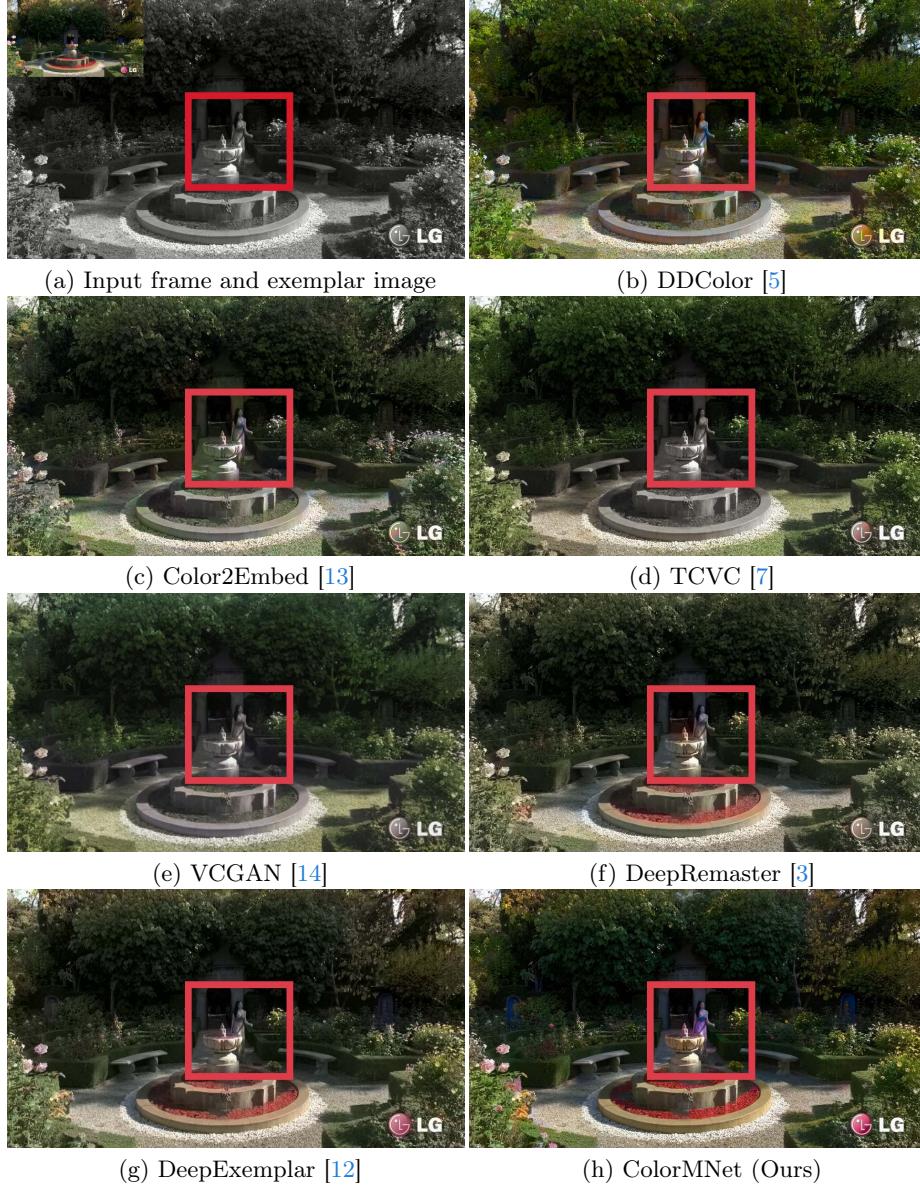


Fig. 11: Colorization results on clip 001 from the NVCC2023 validation dataset [4]. The results shown in (b) and (c) still contain significant color-bleeding artifacts. In contrast, our proposed method generates a better-colorized frame, where the colors of the woman and the leaves are better restored.



Fig. 12: Colorization results on clip 014 from the NVCC2023 validation dataset [4]. The results shown in (b) and (c) still contain significant color-bleeding artifacts. In contrast, our proposed method generates a vivid frame in (h) that is not only more colorful compared with (d-g) but also faithful to the exemplar image in (a).

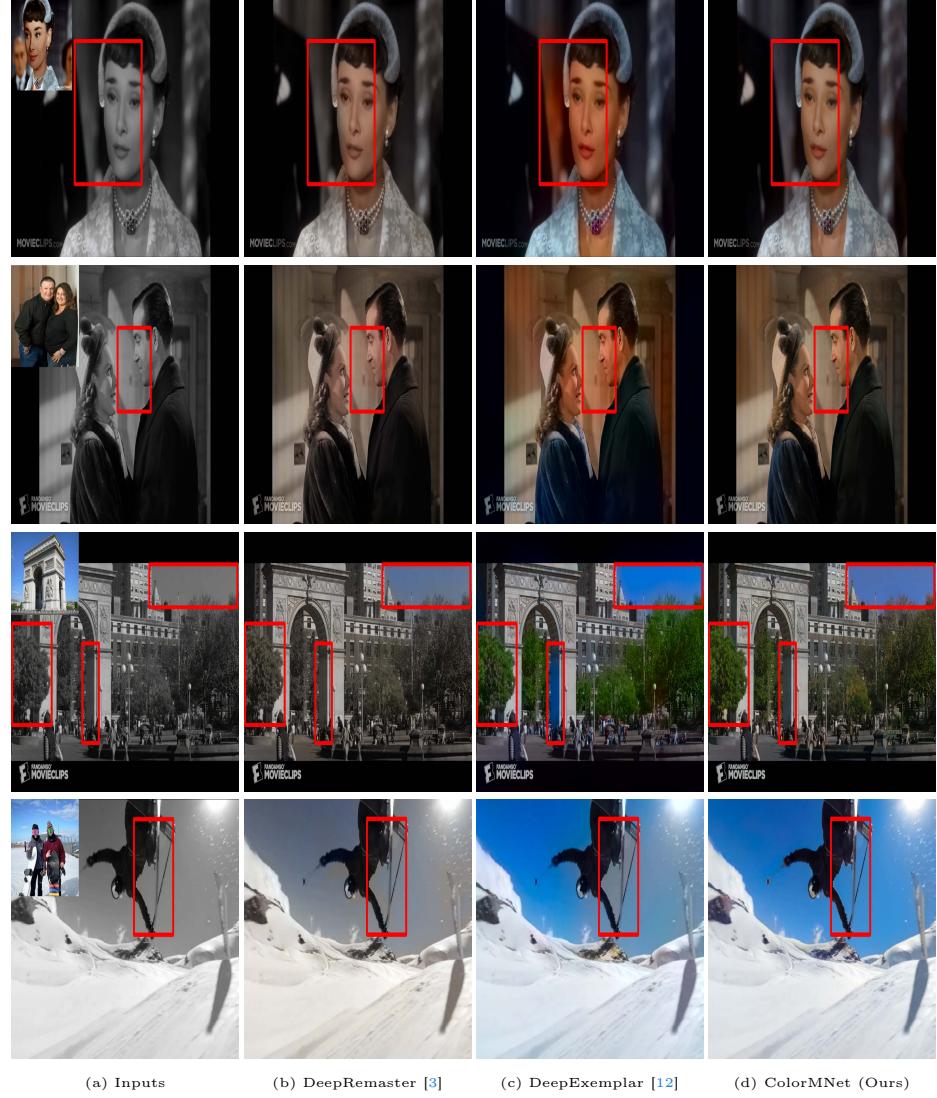


Fig. 13: Colorization results on real-world videos. From top to bottom are respectively the film clip from *Roman Holiday* (1953), the film clip from *Miracle on 34th Street* (1947), the film clip from *Manhattan* (1979) and a real-world video collected from the internet. We obtain the exemplars by searching the internet to find the most visually similar images to the input video frames. DeepRemaster [3] cannot generate vivid frames. The results shown in (c) generated by DeepExemplar [12] still contain significant color-bleeding artifacts (the wall of the building and the skiing man) and cannot maintain faithfulness to the given exemplar images (over-saturated colors on the both the face of the man and the face of the woman). In contrast, our proposed method generates vivid and realistic frames.