SVCNet: Scribble-based Video Colorization Network with Temporal Aggregation

Yuzhi Zhao, *Graduate Student Member, IEEE,* Lai-Man Po, *Senior Member, IEEE,*
Kangcheng Liu, *Member, IEEE,* Xuehui Wang, *Student Member, IEEE*, Wing-Yin Yu, *Student Member, IEEE*,
Pengfei Xian, *Student Member, IEEE*, Yujia Zhang, Mengyang Liu

***Abstract*—In this paper, we propose a scribble-based video colorization network with temporal aggregation called SVCNet. It can colorize monochrome videos based on different user-given color scribbles. It addresses three common issues in the scribble-based video colorization area: colorization vividness, temporal consistency, and color bleeding. To improve the colorization quality and strengthen the temporal consistency, we adopt two sequential sub-networks in SVCNet for precise colorization and temporal smoothing, respectively. The first stage includes a pyramid feature encoder to incorporate color scribbles with a grayscale frame, and a semantic feature encoder to extract semantics. The second stage finetunes the output from the first stage by aggregating the information of neighboring colorized frames (as short-range connections) and the first colorized frame (as a long-range connection). To alleviate the color bleeding artifacts, we learn video colorization and segmentation simultaneously. Furthermore, we set the majority of operations on a fixed small image resolution and use a Super-resolution Module at the tail of SVCNet to recover original sizes. It allows the SVCNet to fit different image resolutions at the inference. Finally, we evaluate the proposed SVCNet on DAVIS and Videvo benchmarks. The experimental results demonstrate that SVCNet produces both higher-quality and more temporally consistent videos than other well-known video colorization approaches. The codes and models can be found at https://github.com/zhaoyuzhi/SVCNet.**

***Index Terms*—Video Colorization, Scribble-based Colorization, Temporal Aggression, Segmentation.**

I. Introduction

Video colorization is the process of attaching plausible colors to monochrome videos. Restricted by imaging technology, many old films are preserved in black-and-white format. It is highly desirable for people to watch colorful videos. Recently, deep neural networks have achieved great improvements in both video restoration and colorization areas. Therefore, recovering realistic and colorful videos with deep neural networks becomes plausible.

The main difficulties for video colorization are *colorization vividness* and *temporal consistency* of sequential frames. Besides, *color bleeding* (i.e., the spreading of colors beyond the object boundary) is another challenge. There are many solutions for existing methods to these problems, which typically fall into one of these four categories:

1) Image colorization and temporal smoothing [1]–[4];
2) Image colorization and color propagation [5]–[8];
3) Fully-automatic video colorization [9]–[11];
4) Exemplar-based video colorization [12]–[14].

The first three categories are not based on additional guidance such as exemplar images and color scribbles. To learn the grayscale to color mapping, they normally adopt large training sets like ImageNet [15] to learn rich data priors. The differences between the three categories are obvious. Since category 1) relies on pre-trained image colorization methods and only finetunes the single image colorization results, video continuity cannot be ensured. Category 2) is similar to category 1) but only uses the first colorized frame. It depends on the long-range connection too much and easily ignores the characteristics of every single frame. Category 3) performs better than 1) and 2) since it jointly learns colorization and temporal smoothing. However, they are difficult to predict realistic color embeddings since the grayscale format losses too much information compared with the color format (e.g., *RGB*, *YUV*, and *CIE Lab*). Furthermore, they may sacrifice colorfulness due to temporal constraints. To improve colorization quality, category 4) induces an exemplar image to guide the video colorization. Though it produces more colorful videos than other methods, it requires a relatively accurate exemplar image similar to the color version of the monochrome input.

To achieve higher video colorization quality than existing methods and minimize color bleeding artifacts, we propose the first scribble-based video colorization framework called SVCNet, where the data flow is illustrated in Figure 1. Compared with previous solutions, there are four improvements:

1) Improved colorization vividness: SVCNet includes two stages for precise colorization (CPNet) and temporal smoothing (SSNet), respectively. The CPNet is a multi-input-multi-output architecture that achieves better colorization quality;

2) Strengthened temporal consistency: SVCNet includes both bidirectional projection and long-range connection, which effectively aggregate the temporal information of the video;

3) Reduced color bleeding artifacts: SVCNet learns an additional auxiliary segmentation task to minimize color bleeding artifacts besides performing video colorization;

4) Reduced manual work: SVCNet only needs sparse color scribbles instead of an accurate exemplar image.

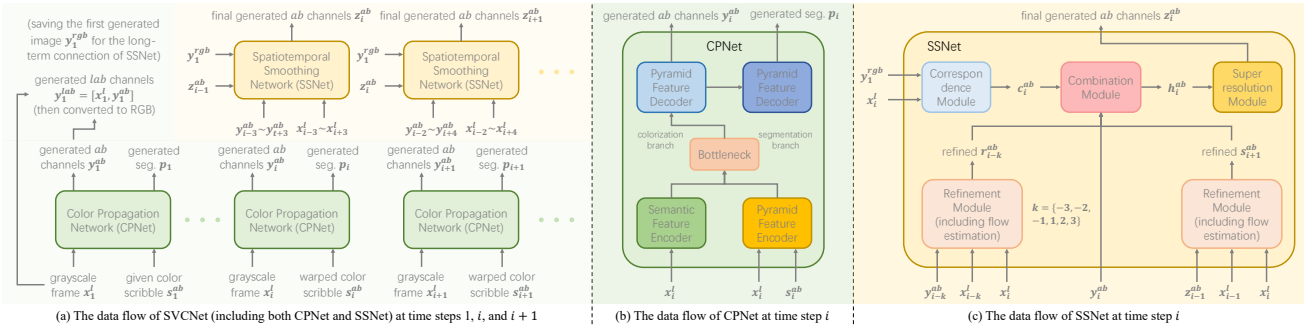
Manuscript under review. *(Corresponding author: Yuzhi Zhao.)*

Y. Zhao, L.-M. Po, W.-Y. Yu, and P. Xian are with the Department of Electronic Engineering, City University of Hong Kong, Hong Kong, China (e-mail: yzzhao2-c@my.cityu.edu.hk; eelmpo@cityu.edu.hk; wingyinyu8-c@my.cityu.edu.hk; xian.pf@my.cityu.edu.hk).

K. Liu is with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. (email: kangcheng.liu@ntu.edu.sg).

X. Wang is with the Artificial Intelligence Institute, Shanghai Jiao Tong University, Shanghai, China (e-mail: wangxuehui@sjtu.edu.cn).

Y. Zhang and M. Liu are with Tencent Video, Tencent Holdings Ltd, Shenzhen, China (e-mail: yujiazhang@tencent.com; myleonliu@tencent.com).



(a) The data flow of SVCNet (including both CPNet and SSNet) at time steps 1, i, and $i+1$ (b) The data flow of CPNet at time step i (c) The data flow of SSNet at time step i

Fig. 1. Illustration of the data flows of (a) SVCNet (including CPNet and SSNet), (b) CPNet, and (c) SSNet. Users only need to provide color scribbles for the first frame s_1^{ab}. The following color scribbles s_2^{ab}-s_t^{ab} are obtained by warping s_1^{ab} using the forward optical flows.

SVCNet includes two sub-networks: color propagation sub-network (CPNet) and spatiotemporal smoothing sub-network (SSNet), respectively. Its data flow is shown in Figure 1. Firstly, CPNet is a multi-input-multi-output architecture, as shown in Figure 1 (b). It includes two encoders to extract semantics from the grayscale input and combine the information of both inputs, which performs precise scribble-color propagation and ensures good colorization quality when there are even no color scribbles. In addition, it has two decoder branches, where the colorization branch outputs color embeddings and the other outputs the corresponding segmentation map. The segmentation branch assists the colorization branch through backward propagation at the training. It helps the CPNet reduce the color bleeding artifacts since it pushes the network to separate clear boundaries. Secondly, SSNet refines every colorized frame of the CPNet, as shown in Figure 1 (c). On one hand, it performs the bidirectional projection based on previous, current, and leading frames by a Refinement Module, which serve as short-range connections. On the other hand, it extracts and aligns the first colorized frame as the long-range connection by a Correspondence Module. After that, a Combination Module aggregates all the information. Therefore, SSNet colorizes videos with satisfactory temporal consistency. We notice color embeddings (i.e., *ab* channels in the *CIE Lab* color space) are much sparser than edges. Based on this and also inspired by [16]–[19], we set all the aforementioned operations on a small and fixed image resolution and use a Super-resolution Module at the end of the SSNet to recover the original image resolution. Therefore, SVCNet fits different image resolutions at the inference. Finally, compared with exemplar-based video colorization, the proposed framework only needs sparse color scribbles input. Users do not need to select a proper exemplar image while only need to define the desired colors in some specific pixels.

We train and evaluate the SVCNet on both DAVIS [20] and Videvo [21] datasets. Extensive experiments demonstrate that SVCNet performs better than state-of-the-art video colorization methods. We also show that the CPNet achieves state-of-the-art scribble-based image colorization performance with fewer color bleeding artifacts. The main contributions of this paper are as follows:

1) We propose the first scribble-based video colorization framework called SVCNet, which includes two stages for color propagation and spatiotemporal smoothing, respectively. We set the most operations on a small and fixed image resolution to reduce the computational costs for producing videos with different large resolutions;

2) We propose a temporal aggregation method for video colorization including both short- and long-range connections;

3) We adopt a segmentation loss to address color bleeding artifacts in the video colorization area. Also, we generate saliency maps as pseudo-binary segmentation maps when there are no labeled segmentation maps in the datasets.

II. Related Work

Image Colorization. Image colorization learns to reconstruct color embeddings from corresponding grayscale images. It can be categorized into reference-based colorization (e.g., scribble-based colorization [22]–[33], exemplar-based colorization [5], [6], [12], [13], [34]–[46], text-based colorization [47]–[49]) and fully-automatic colorization [18], [50]–[66]. Reference-based methods require additional user inputs that contain information relevant to the desired colors. The colorization systems use such information to assign possible colors to grayscale input images. Specifically, scribble-based methods propagate user-given color scribbles to the rest of the image. Exemplar-based methods attach colors from exemplar images to grayscale images. Text-based methods translate the information from words or languages into colors. Recently, deep neural networks improve colorization performance thanks to their superior feature representation ability. Built upon them, fully-automatic methods directly learn end-to-end colorization without any additional information on large datasets. To further improve the colorization quality, some specific designs have been used such as hyper-column [51], pre-trained backbones [51]–[53], [57], [59], multi-task learning [53], [56]–[59], and auxiliary loss functions [54].

Video Colorization. There are mainly four categories of existing video colorization methods in terms of data flow, as shown in Figure 2. The descriptions are as follows:

1) Image colorization and temporal smoothing [1], [2], [4], [67], [68] (Figure 2 (a)): Based on the progress of image colorization methods, they added the temporal consistency on single colorized frames by post-processing them;

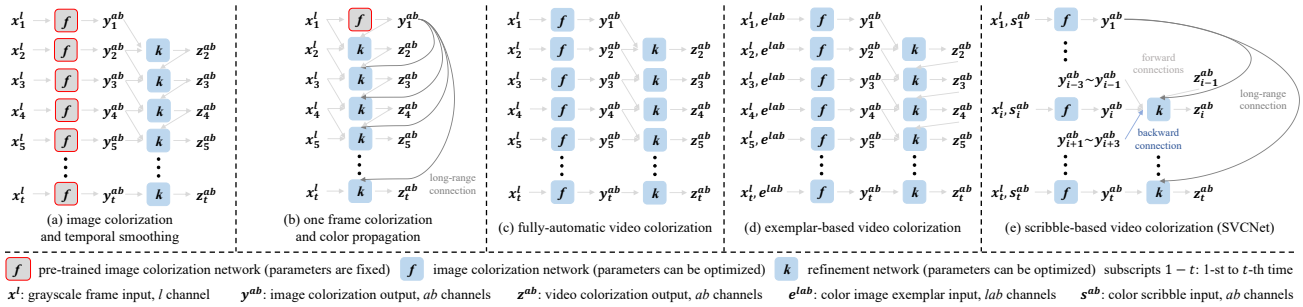


Fig. 2. Illustration of data flows of video colorization methods: (a) image colorization and temporal smoothing [1]–[4]; (b) one frame colorization and color propagation [5]–[8]; (c) fully-automatic video colorization [9]–[11]; (d) exemplar-based colorization [12]–[14]; (e) scribble-based colorization (SVCNet), where only the first color scribble s_1^{ab} is given by the user. The following color scribbles s_2^{ab}, s_3^{ab}, ... , s_t^{ab} are obtained by warping the first color scribble s_1^{ab} using forward optical flows $O_{1\to2}$, $O_{2\to3}$, ... , $O_{t-1\to t}$.

2) Image colorization (e.g., the first frame) and color propagation [5]–[8] (Figure 2 (b)): They use one colorized frame to start the colorization. The colors and the correspondence of the following frames are learned sequentially;

3) Fully-automatic video colorization [9]–[11] (Figure 2 (c)): They learn the video colorization and temporal correspondence jointly on large video datasets by neural networks instead of learning it individually as categories 1) and 2);

4) Exemplar-based video colorization [12]–[14] (Figure 2 (d)): They propagate the colors from the exemplar image to the monochrome frames in a video.

Normally, the performance of both categories 1) and 2) are not satisfactory since the optimization of image colorization and temporal smoothing is separated. They require a powerful pre-trained image colorization algorithm, but normally the colorized videos are temporally not consistent. To address the issue, the other two categories combine image colorization and temporal smoothing together and learn them jointly. For instance, Lei *et al.* [9] proposed a two-stage multi-modal video colorization framework. Kouzouglidis *et al.* [10] adopted 3D convolution as the basic operator. Zhao *et al.* [11] used a global feature extractor and a placeholder feature extractor in the generator. Though the methods generate colorful videos automatically, their results are still not vivid enough. To improve the colorization quality, category 4) adopted an additional exemplar guidance image, e.g., Zhang *et al.* [12] aligned the exemplar with grayscale frames by a Nonlocal network and then fuse them by a ColorNet. It requires a high-quality exemplar image to obtain satisfactory results. To further ease the exemplar selection, we propose the first scribble-based video colorization method called SVCNet. The data flow of SVCNet is shown in Figure 2 (e).

Scribble-based Colorization. The scribble-based colorization aims to propagate colors from user-given color scribbles to monochrome images. Common propagation schemes use local correspondences [22], edges [23], or luminance-weighted chrominance blending [24]. However, these methods focused on local relations and failed to colorize the pixels far to color scribbles. To model the long-range connection, Xu *et al.* [27] proposed an affinity-based image editing scheme and Chen *et al.* [26] learned the mapping in feature space. However, the results are still highly related to the number or the location of given color scribbles. Moreover, the color bleeding effect is obvious when given color scribbles are close to the edges of objects. Recently, Zhang *et al.* [30] used a neural network to extract the semantics and achieved better performance. In this paper, we further extend the scribble-based colorization to videos by the SVCNet framework.

Saliency Detection. It aims to localize the potential perceptual significant regions of the image by "saliency map". The early saliency detection methods were based on hand-crafted features such as color variation [69], boundaries [70], and super-pixel [71], which predicted credible boundaries but not accurate structures of salient objects. Recent deep-learning-based methods generalized the saliency detection to diverse images and adopted different architectures such as recurrent network [72], encoder-decoder [73]–[76], and feature pyramid network [77]–[80] to fuse details and high-level semantics.

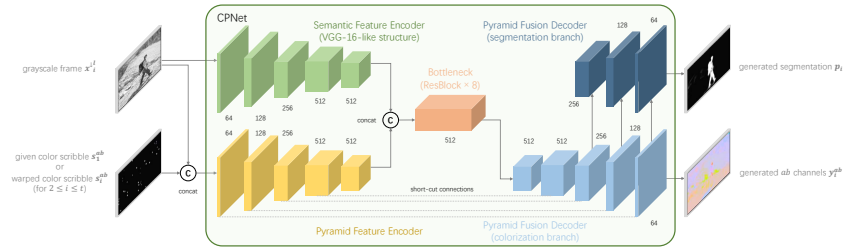
Semantic Segmentation. It aims to localize different objects in an image. The pioneer deep-learning-based method FCN [81] used deconvolution and fusion of pooling layers. To enhance feature representation and context information, PSPNet [82] adopted a pyramid pooling module and DeepLab [83] used different rates of dilated convolution. Bulti upon it, Chen *et al.* [84] combined the ASPPM with encoder-decoder architecture in DeepLab to let the network capture features in both cross and intra layers. Yang *et al.* [85] proposed a DenseASPPM to assemble different dilated branches. Recently, visual recognition in other data modalities such as 3D segmentation [86]–[88] has achieved decent performance.

III. Methodology

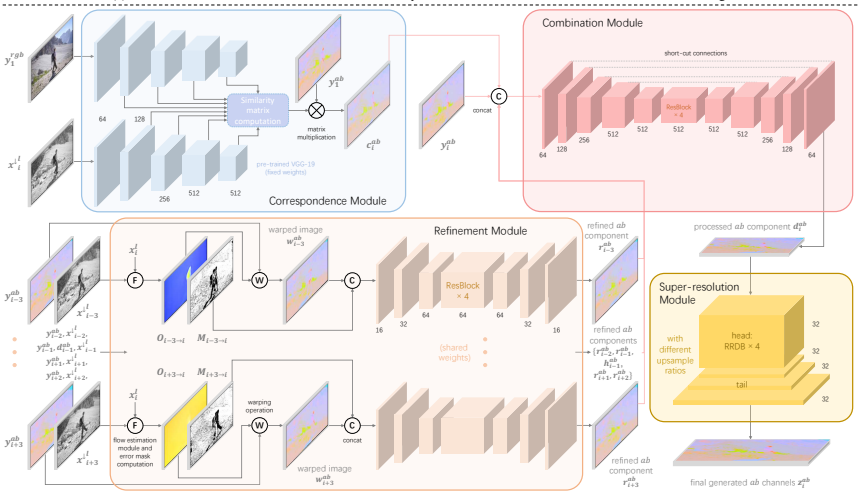
A. Problem Formulation

Given continuous grayscale frames, we aim to generate realistic color embeddings based on user-given color scribbles. To assist the colorization, we also generate corresponding segmentation maps to minimize color bleeding artifacts. We formulate the problem as maximizing a posteriori of color embeddings and segmentation maps conditioned on the grayscale frames $x_1^l, x_2^l, ..., x_t^l$, user-given color scribbles s_1^{ab}, and network parameters Θ of the SVCNet:

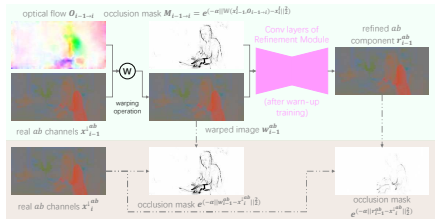
$$\Theta^* = \arg\max_{\Theta} p(x_{1:t}^{ab}, q_{1:t} | x_{1:t}^l, s_1^{ab}, \Theta), \tag{1}$$



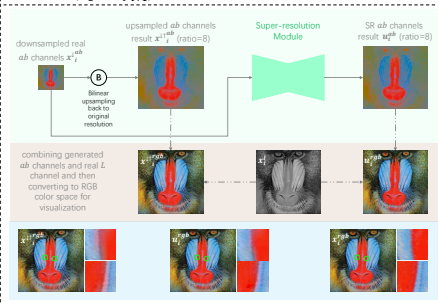
(a) Illustration of the CPNet architecture. The detailed layers of different modules of the CPNet are shown in the figure.



(b) Illustration of the SSNet architecture. The detailed layers and channel numbers of different modules of the SSNet are shown in the figure.



(c) Illustration of training data flow and warm-up training results of the Refinement Module. The forward optical flow $\boldsymbol{O_{i-1\to i}}$ is computed by PWCNet and between two consecutive grayscale frames $\boldsymbol{x_{i-1}^l}$ and $\boldsymbol{x_i^l}$. The occlusion mask $\boldsymbol{M_{i-1\to i}}$ is computed between two consecutive grayscale frames $\boldsymbol{x_{i-1}^l}$ and $\boldsymbol{x_i^l}$. It is clear that the occlusion mask of the refined result and real ab channels is cleaner than it before the refinement. The backward refinement (e.g., for $\boldsymbol{y_{i+1}^{ab}}$) is similar to the forward refinement.



(d) Illustration of training data flow and warm-up training results of the Super-resolution Module. The low-resolution image $\boldsymbol{x_i^{ab}}$ is obtained by downsampling real ab channels $\boldsymbol{x_i^{ab}}$, where the downsampling operator is the bilinear interpolation. At the right, we highlight two patches from bilinear upsampled result $\boldsymbol{x^{\downarrow\uparrow}{}_i^{rgb}}$, super resolution result $\boldsymbol{u_i^{rgb}}$, and real image $\boldsymbol{x_i^{rgb}}$ in the RGB color space. It is clear that $\boldsymbol{u_i^{rgb}}$ has much less artifacts and color bleeding effects than $\boldsymbol{x^{\downarrow\uparrow}{}_i^{rgb}}$ when the upsampling ratio equals 8.

Fig. 3. Illustration of the detailed architectures for (a) CPNet and (b) SSNet. The special blocks are annotated while the other blocks are normal convolutional layers. The number of channels is annotated beside the blocks. Illustration of the training data flow and warm-up training results of (c) SSNet Refinement Module and (d) SSNet Super-resolution Module.

where $x_{1:t}^l$ and $x_{1:t}^{ab}$ are the grayscale component and color component in the *CIE Lab* color space from the 1-st frame to the t-th frame. $q_{1:t}$ are ground truth segmentation maps. Θ^* are theoretically optimal SVCNet parameters. To optimize the network, we further formulate the loss function L on both color embeddings and segmentation maps:

$$\underset{SVCNet(.;\Theta)}{minimize} \sum_{i=1}^{t}(L(z_i^{ab}, x_i^{ab}) + L(p_i, q_i)), \tag{2}$$

where z_i^{ab} and p_i are the outputs generated by the SVCNet at time step i. x_i^{ab} and q_i are ground truth color embeddings and segmentation maps at time step i. $SVCNet(.;\Theta)$ denotes the SVCNet with network parameter Θ.

B. SVCNet Architecture

The SVCNet consists of two sub-networks: color propagation sub-network (CPNet) and spatiotemporal smoothing sub-network (SSNet), as shown in Figures 1 and 3. Below we present every sub-network and module.

CPNet. CPNet performs scribble-based image colorization, which includes two encoders and two decoders, as shown in Figure 3 (a). To merge the information of grayscale input and color scribble guidance, we adopt a pyramid feature encoder. Considering that the color scribbles for far time steps are obtained by warping the first color scribble may vanish, we strengthen the colorization quality by the other semantic feature encoder, which is a pre-trained network. The resulting features are concatenated and processed by a bottleneck for fusion. Afterward, the pyramid feature decoder produces a color embedding from the output of bottleneck and short-cut connections of the pyramid feature encoder [89]. In addition, the last three decoder layers are fed into a segmentation branch to produce corresponding segmentation maps. At the training, the segmentation branch helps revise the weights of the colorization branch through backpropagation.

SSNet. SSNet post-processes the output of CPNet. It enhances the temporal consistency of generated frames and includes four modules, as shown in Figure 3 (b).

Refinement Module. It is designed for processing short-range correlations, which are modeled by both previous frames $(y_{i-3}^{ab}, y_{i-2}^{ab}, y_{i-1}^{ab}, d_{i-1}^{ab})$ and leading frames $(y_{i+1}^{ab}, y_{i+2}^{ab}, y_{i+3}^{ab})$ on time step i. To match their locations with the current frame, we first warp them to the position of the current frame with optical flows, which are computed by a pre-trained PWC-Net [90] on grayscale frames. Then, we use the Refinement Module to post-process them in order to minimize the warping artifacts caused by occlusions and motion boundaries. To represent such regions, we compute the occlusion mask for warped previous and leading frames, which are defined as:

$$M_{j\to i} = exp(-\alpha||\mathbb{W}(x^{\downarrow l}{}_j, O_{j\to i}) - x^{\downarrow l}{}_i||_2^2), \tag{3}$$

where $M_{j\to i}$ and $O_{j\to i}$ denote the occlusion mask and the optical flow from time step j to i, respectively. $x^{\downarrow l}{}_j$ and $x^{\downarrow l}{}_i$ are downsampled real grayscale images at time steps j and

i, respectively. $\mathbb{W}(*)$ is the warping operation. α is a hyper-parameter that controls the sensitivity of the occlusion mask. The forwarding procedure of Refinement Module (RM) can be represented as:

$$r_j^{ab} = \mathrm{RM}(\mathbb{W}(y_j^{ab}, O_{j\to i}), M_{j\to i}), \tag{4}$$

where we set $j = (i-3, i-2, i-1, i+1, i+2, i+3)$. Also, we refine the previous SSNet low-resolution output d_{i-1}^{ab}, which is the output of the Combination Module, as follows:

$$h_{i-1}^{ab} = \mathrm{RM}(\mathbb{W}(d_{i-1}^{ab}, O_{i-1\to i}), M_{i-1\to i}). \tag{5}$$

Correspondence Module. The long-range connection between the first colorized frame and the current frame is modeled by the Correspondence Module. We do not use the optical flow since there often exist large motions between them. On the contrary, we compute the similarity matrix between them and use it to warp the features of the first generated color embeddings. Firstly, like in [12], we build the feature pyramid of RGB-formatted first frame y_1^{rgb} and current grayscale frame $x^{\downarrow l}_i$ ($y_1^{rgb}, x^{\downarrow l}_i \in \mathbb{R}^{H\times W\times C}$) by extracting their features of layers $conv_{2\_2}$, $conv_{3\_2}$, $conv_{4\_2}$, and $conv_{5\_2}$ of a pre-trained VGG-19 network [91]. All the features are upsampled to the same resolution as $conv_{2\_2}$ (i.e., $\frac{1}{2}H \times \frac{1}{2}W$) and then concatenated together. The resulting features for y_1^{rgb} and $x^{\downarrow l}_i$ are denoted as F_1 and F_i, respectively. Secondly, the similarity matrix $S_{1\leftrightarrow i} \in \mathbb{R}^{\frac{1}{4}HW\times\frac{1}{4}HW}$ is computed as:

$$S_{1\leftrightarrow i} = \frac{F_1 - \mu(F_1)}{||F_1 - \mu(F_1)||_2} \cdot \frac{F_i - \mu(F_i)}{||F_i - \mu(F_i)||_2}, \tag{6}$$

where $\mu(*)$ denotes the mean operation. Then, we use the similarity matrix to warp y_1^{ab} [12] as follows:

$$c_i^{ab} = (\sum_m \underset{h}{softmax}(\tau \cdot S_{1\leftrightarrow t}(:, m)) \cdot y^{\downarrow ab}_1)^{\uparrow}, \tag{7}$$

where c_i^{ab} is the warped result of the first generated color embeddings y_1^{ab}. τ is the temperature parameter. There is a downsampling operation since we need to match the resolution of the similarity map and the color embeddings.

Combination Module. Next, the Combination Module aggregates the information from outputs of the Refinement Module ($r_{i-3}^{ab}, r_{i-2}^{ab}, r_{i-1}^{ab}, h_{i-1}^{ab}, r_{i+1}^{ab}, r_{i+2}^{ab}, r_{i+3}^{ab}$), Correspondence Module (c_i^{ab}), and the current generated color embeddings from the CPNet (y_i^{ab}) by a U-Net-like architecture [89]. The output of the Combination Module is denoted as d_i^{ab}.

Super-resolution Module. Finally, since previous operations run on a fixed small resolution, the Super-resolution Module recovers the frames with original sizes at inference from d_i^{ab}. For different high resolutions, we use the same feature extraction head but different tails with multiple upsampling ratios. The final output is denoted as z_i^{ab}.

C. Training Strategy

Directly optimizing the large SVCNet without any initialization easily encounters the gradient exploding issue. To stabilize the training, we propose the warm-up pre-training for CPNet and some modules of SSNet, respectively. After that, we train the full SVCNet on video datasets. The details are as follows:

1) CPNet warm-up pre-training: We pre-train the CPNet on ImageNet dataset [15], which provides much more modes and scenes than video datasets. Since the ImageNet dataset does not provide segmentation maps, we generate saliency maps as ground truth by [78]. We then finetune the CPNet on single frames from video datasets (DAVIS [20] and Videvo [21]). It makes CPNet fit the sizes of video frames better. Similarly, we generate saliency maps as pseudo segmentation maps for the Videvo dataset by [78].

2) SSNet warm-up pre-training: We conduct the self-supervised learning for the Refinement Module and the Super-resolution Module on two video datasets: DAVIS [20] and Videvo [21]. Firstly, for a frame x_i^{ab} in a video, we stochastically warp the previous frame x_{i-1}^{ab} or leading frame x_{i+1}^{ab} to the position of x_t^{ab} as the input for the Refinement Module. We make it reconstruct itself and train it with an $L1$ loss. Secondly, we downsample a frame of the original resolution with different ratios (2, 4, and 8) as the input for the Super-resolution Module. We then let the Super-resolution Module reconstruct itself and train it with an $L1$ loss. The warm-up training is conducted only on *ab* color components. It ends when the loss is small and stable enough. The results are shown in Figure 3 (c) and (d). The Refinement Module can reduce the artifacts in the occluded regions. The Super-resolution Module can minimize the color bleeding artifacts especially when the upsampling ratio is large (e.g., 8).

3) joint training stage: We optimize the full SVCNet (CPNet and SSNet) based on all the warm-up training weights. We randomly select 7 continuous frames in a video and then flip the first 6 frames as a batch at the training. In what follows, we will introduce the loss functions for the "CPNet warm-up pre-training" and "joint training stage".

D. Loss Function

When training CPNet, we apply a colorization loss and a segmentation loss for the outputs of the two pyramid fusion decoder branches, respectively:

$$L_c^{CP} = \mathbb{E}[||y_i^{ab} - x^{\downarrow ab}_i||_1], \tag{8}$$

$$L_s^{CP} = \mathbb{E}[||p_i - q_i||_1], \tag{9}$$

where y_i^{ab} and p_i are the outputs of two branches. $x^{\downarrow ab}_i$ and q_i are the corresponding ground truth. Then, the total loss function for the CPNet warm-up training is defined as:

$$L^{CP} = L_c^{CP} + \lambda_s L_s^{CP}, \tag{10}$$

where λ_s is the trade-off parameter.

For the joint training, we maintain the CPNet loss function and add the SSNet loss function. Specifically, we adopt colorization losses for the output of the Combination Module (d_i^{ab}) and the output of the Super-resolution Module (z_i^{ab}), respectively. They can be represented as:

$$L^{SS} = \mathbb{E}[||d_i^{ab} - x^{\downarrow ab}_i||_1] + \mathbb{E}[||z_i^{ab} - x_i^{ab}||_1], \tag{11}$$

where $x^{\downarrow ab}_i$ and x_i^{ab} are ground truth at low resolution and original resolution, respectively.

The full loss function of the joint training stage is the sum:

$$L^{joint} = L^{CP} + L^{SS}. \tag{12}$$

TABLE I
CONCLUSION OF OPTIMIZATION DETAILS FOR DIFFERENT TRAINING STAGES.

Training stage	Trained network	Loss	Training set	Total iterations	Initial learning rate (LR)	LR halved iteration
1) CPNet warm-up	CPNet	L^{CP}	ImageNet	800,740	1×10^{-4}	400,370
pre-training	CPNet	L^{CP}	DAVIS+Videvo	200,600	5×10^{-5}	100,300
2) SSNet warm-up	Refinement Module	$L1$	DAVIS+Videvo	312,000	1×10^{-4}	156,000
pre-training	Super-resolution Module	$L1$	DAVIS+Videvo	296,200	1×10^{-4}	148,100
3) joint training stage	CPNet+SSNet	L^{joint}	DAVIS+Videvo	312,000	CPNet:1×10^{-6}; SSNet:5×10^{-5}	156,000

IV. EXPERIMENT

A. Implementation Details

Training Dataset. We use the entire ImageNet dataset for the CPNet warm-up pre-training. It includes 1.3 million images with 1000 categories. We use DAVIS and Videvo datasets for other training stages. They include 156 short video clips with 29620 frames. The DAVIS dataset has labeled binary segmentation maps. For ImageNet and Videvo datasets, we generate saliency maps as pseudo-segmentation maps. As for pre-processing, ImageNet images are resized to the resolution of 256×256. The video frames from DAVIS and Videvo datasets are resized to 256×448 as the input of the SVCNet and 512×896 as the ground truth (i.e., the upsampling ratio for the Super-resolution Module is set to 2 at the joint training stage). All images are normalized to the range of [0, 1].

Color Scribble. We randomly select color scribbles from *ab* channels of ground truth images. There is a half probability to use valid color scribbles in the training and the number of color scribbles ranges from 1 to 40. At the joint training stage, color scribbles are only provided for the first frame in a batch. The following color scribbles are obtained by warping the previous one with forward optical flows.

Optimization. The optimization details are concluded in Table I, where we list trained networks, loss functions, training sets, total training iterations, initial learning rates (LRs), and the specific iterations when learning rates halved, respectively, in every training stage. At the start of the joint training stage, we initialize both CPNet and SSNet with warm-up pre-training weights. For the remaining layers or blocks, we initialize them by [92]. The batch size for the warm-up pre-training stages is 4 and it is 1 for the joint training stage per GPU. We use the Adam optimizer [93] with β_1=0.5, β_2=0.999. The trade-off parameter λ_s is set to 0.1. Both the mask parameter α and temperature parameter τ are 200. We implement the SVCNet with the PyTorch framework. We train it on 8 NVIDIA V100 GPUs and 8 NVIDIA Titan Xp GPUs. Considering the parallel training, it takes approximately 10 days to complete the warm-up pre-training and another 6 days to complete the joint training.

Network Architecture. The SVCNet architecture is shown in Figure 3, where we emphasize special blocks such as short-cut connections [89], residual block (ResBlock) [94], and residual-in-residual dense block (RRDB) [95]. We also emphasize the number of channels. We use LeakyReLU [96] as the activation function except for the first and second layers. We use instance normalization [97] only in the Refinement Module.

B. Quantitative Metrics

Generation Quality. We adopt PSNR and SSIM [98] to calculate pixel-level accuracy and structural similarity between generated results and ground truth, respectively. For the scribble-based colorization task, some ground truth scribbles are given in the validation stage and the colorization becomes a specific task. Therefore, PSNR and SSIM are proper metrics to evaluate the generation quality.

Segmentation Performance. We use HRNetV2 + OCR [99], [100] to calculate the mean intersection over union (mIoU) of the generated frames on DAVIS semantic segmentation validation dataset [20]. If a method obtains a higher mIoU value, it may have less probability to encounter the color bleeding issue since the segmentation algorithm can better separate key objects from the colorized image of this method.

Human Preference. We conduct a human preference study on video colorization results from different methods. There are 10 videos randomly selected from DAVIS and Videvo validation sets for users to compare. For each video, the human observer needs to select the best result based on temporal consistency and color vividness. There are 10 human observers in the experiment and they can watch the videos many times.

C. Video Colorization Experiments

Experiment Setting. We compare the video colorization performance of SVCNet and other recent works with similar targets, which can be categorized into four pipelines:

1) Image colorization and temporal smoothing (as shown in Figure 2 (a)): CIC [52], LTBC [53], ChromaGAN [57], and IAC [58] are used as image colorization algorithms; BTC [2] and DVP [4] are used as temporal smoothing networks;

2) Image colorization and color propagation (as shown in Figure 2 (b)): DEVC [12] is used to propagate the first colorized frame to the following frames;

3) Fully-automatic video colorization (as shown in Figure 2 (c)): FAVC [9], 3DVC [10], and VCGAN [11];

4) Scribble-based image colorization and temporal smoothing/color propagation: RUIC [30] is used as the image scribble-based image colorization method; DVP [4] and DEVC [12] are used as post-processing methods. Note that, RUIC is finetuned on video datasets.

We perform experiments on DAVIS [20] and Videvo [21] validation sets. There are overall 50 video clips and each of them contains approximately 100 frames. In the experiment, we use 40 color scribbles for scribble-based methods, which are randomly extracted from the first frame. The following scribbles are obtained by warping color scribbles of the first

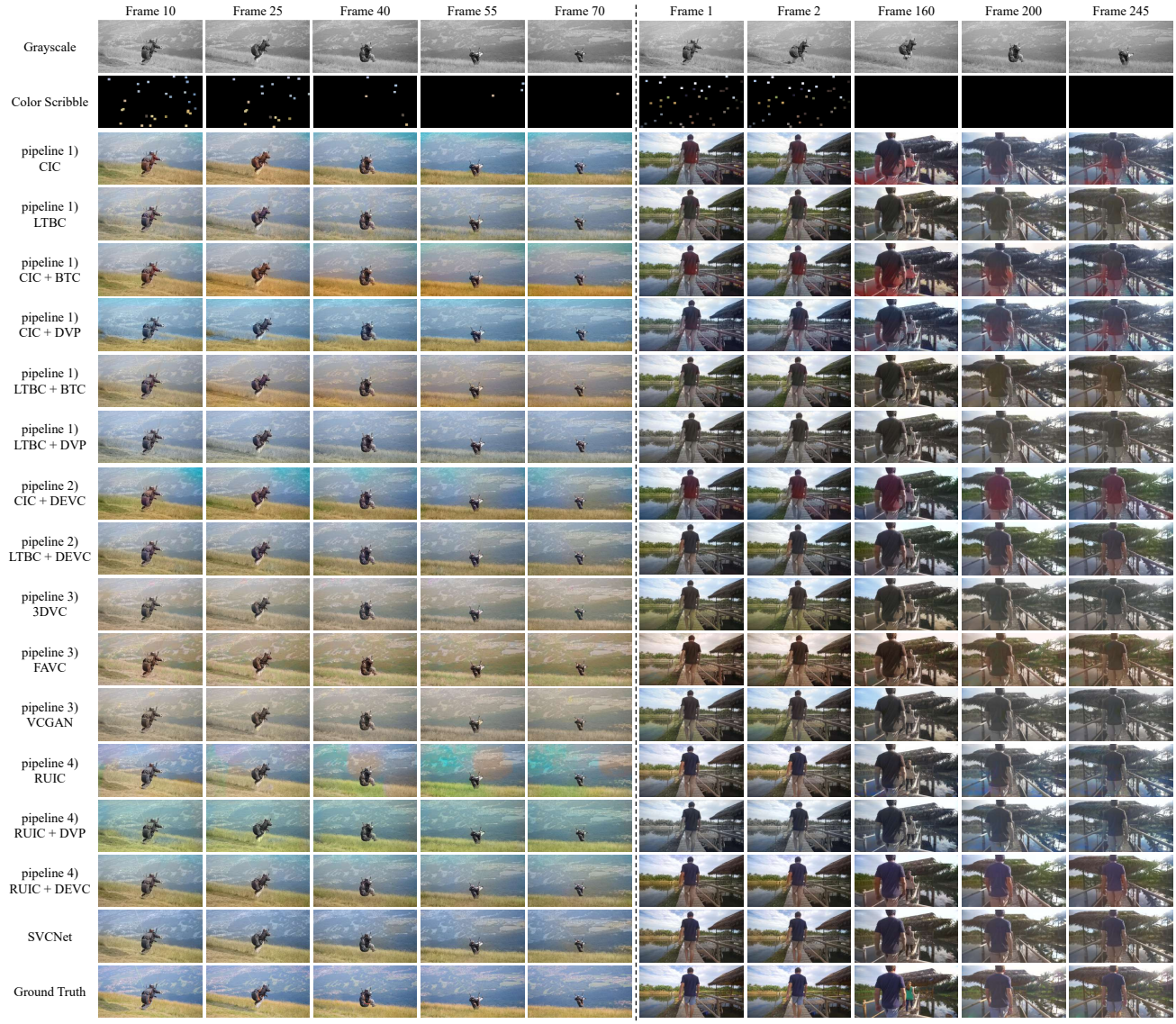


Fig. 4. Illustration of video colorization results of SVCNet and other pipelines. The images are extracted from the "paragliding-launch" sample of the DAVIS dataset and the "YogaHut2" sample of the Videvo dataset, respectively.

frame with forward optical flows. Different pipelines share the same color scribbles. For the SVCNet, we set the up-sampling ratio of the Super-resolution module as 2 to obtain results with a resolution of 512×896 and then resize them to target resolutions of DAVIS and Videvo datasets. The other methods run on the same resolution as validation images. The overall tuning epoch of DVP is set to 25 as default. Note that, FAVC and DVP automatically crop input images to make side lengths a multiple of 32. Therefore, we only use valid regions to calculate quantitative metrics for them. The quantitative results are concluded in Table II and some samples are illustrated in Figures 4 and 5.

Colorization Quality. Firstly, from Figure 4, SVCNet results are more approach to the ground truth than the results of other methods in terms of the color tone and color naturalness. For instance, the results from 3DVC, FAVC, and VCGAN on the "paragliding-launch" sample are less colorful. These methods do not well balance the video colorization vividness and temporal smoothness, i.e., their results are too smooth so they are not colorful enough. Although BTC, DVP, and DEVC can smooth single-colorized frames, the results are still not reasonable. For instance, there are obvious color artifacts in CIC-related results on the "YogaHut2" sample, e.g., the clothes are colorized in red. Since CIC predicts a color distribution for each pixel in an image, the output is not always natural compared with ground truth.

Secondly, there is less color bleeding issue in SVCNet results than in other methods. For instance, the human is colorized to blue for CIC + DVP on the "paragliding-launch" sample since it cannot distinguish the human and the sky. The hand is colorized to green for 3DVC on the "YogaHut2" sample since the neighboring green colors of trees affect the colorization of it. Compared with other methods, SVCNet has much fewer color bleeding artifacts since it uses a segmenta-

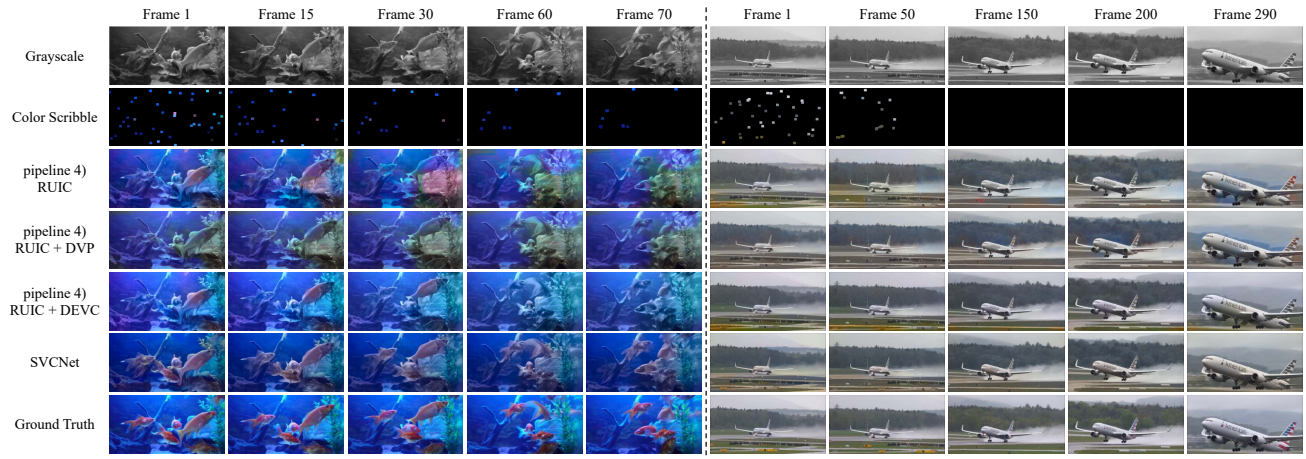


Fig. 5. Illustration of video colorization results of SVCNet and state-of-the-art scribble-based video colorization methods. The images are extracted from the "gold-fish" sample of the DAVIS dataset and the "AircraftTakingOff1" sample of the Videvo dataset, respectively.

TABLE II
COMPARISON OF VIDEO COLORIZATION PIPELINES AND THE PROPOSED SVCNET ON DAVIS AND VIDEVO DATASETS. THE **RED**, **BLUE**, AND GREEN COLORS REPRESENT THE BEST, THE SECOND-BEST, AND THE THIRD-BEST PERFORMANCE, RESPECTIVELY. THE "IC" AND "VC" REPRESENT THE "IMAGE COLORIZATION METHOD" AND "VIDEO COLORIZATION METHOD", RESPECTIVELY.

Method	Type	Color scribble	DAVIS			Videvo	
			PSNR ↑	SSIM ↑	mIoU ↑	PSNR ↑	SSIM ↑
1) CIC [52]	IC	/	22.48	0.9043	0.8419	21.83	0.9024
1) LTBC [53]	IC	/	23.96	0.9171	0.8945	24.69	0.9272
1) ChromaGAN [57]	IC	/	23.77	0.9420	0.8879	23.94	0.9392
1) IAC [58]	IC	/	22.88	0.9422	0.8836	23.99	0.9486
1) CIC + BTC [2], [52]	VC	/	21.50	0.8935	0.8419	21.05	0.8833
1) LTBC + BTC [2], [53]	VC	/	22.47	0.9044	0.8899	22.83	0.9105
1) ChromaGAN + BTC [2], [57]	VC	/	19.91	0.8938	0.8796	16.64	0.8325
1) IAC + BTC [2], [58]	VC	/	18.40	0.8778	0.8698	15.85	0.8209
1) CIC + DVP [4], [52]	VC	/	23.30	0.9351	0.8724	22.23	0.9328
1) LTBC + DVP [4], [53]	VC	/	24.06	0.9425	0.8886	24.75	0.9548
1) ChromaGAN + DVP [4], [57]	VC	/	23.81	0.9444	0.8800	23.97	0.9451
1) IAC + DVP [4], [58]	VC	/	22.91	0.9407	0.8936	23.99	0.9503
2) CIC + DEVC [12], [52]	VC	/	21.64	0.9321	0.8661	21.36	0.9231
2) LTBC + DEVC [12], [53]	VC	/	22.46	0.9397	0.8688	24.03	0.9513
2) ChromaGAN + DEVC [12], [57]	VC	/	22.56	0.9429	0.9026	22.40	0.9386
2) IAC + DEVC [12], [58]	VC	/	22.43	0.9384	0.8712	23.59	0.9479
3) 3DVC [10]	VC	/	23.49	0.9151	0.8948	24.33	0.9231
3) FAVC [9]	VC	/	22.98	0.9055	0.8889	23.47	0.9183
3) VCGAN [11]	VC	/	23.43	0.9133	0.8954	24.73	0.9225
4) RUIC [30]	IC	✓	25.42	0.9456	0.8995	25.02	0.9432
4) RUIC + BTC [2], [30]	VC	✓	21.16	0.8869	0.8900	17.07	0.8279
4) RUIC + DVP [4], [30]	VC	✓	25.82	0.9455	0.9075	24.68	0.9460
4) RUIC + DEVC [12], [30]	VC	✓	24.85	0.9524	0.9002	25.66	0.9583
SVCNet	VC	✓	25.71	0.9565	0.9104	26.30	0.9615

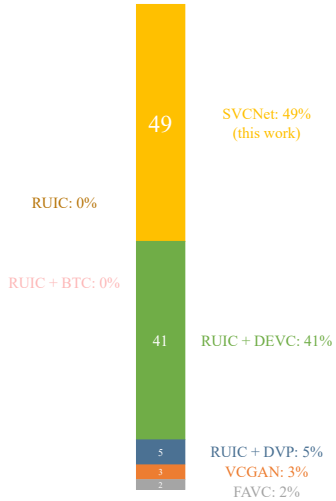


Fig. 6. Human preference study result. The human preference rate for each method is marked in the figure. Different colors denote different methods.

tion branch to alleviate the problem.

Thirdly, SVCNet obtains better video temporal consistency than other methods since it aggregates both short-range connections and the long-range connection. It obtains better temporal consistency than other methods. For pipeline 1), the videos generated by image colorization and temporal smoothing methods are not continuous enough, e.g., CIC + BTC, CIC + DVP, LTBC + BTC, and LTBC + DVP. Since image colorization and temporal smoothing are learned individually, the final outputs are still close to the single-frame colorized results. For instance, the human is colorized to red in frame 160 for the CIC results while it turns to gray for frame 200; but the CIC + BTC cannot alleviate this issue, i.e., the results are not temporal consistent enough. For pipeline 2), the videos are smoother than the results from pipeline 1). However, it exists a similar problem since image colorization methods and DEVC are not jointly trained. For pipeline 3), the video continuity is good compared with other methods, but the results of pipeline 3) are less colorful. For pipeline 4), RUIC + DEVC outperforms the other two methods RUIC and RUIC + DVP. However, it remains the common problem of DEVC-based methods, i.e., results highly rely on the first colorized frame.

Finally, SVCNet better utilizes the given color scribbles than RUIC. Since only color scribbles of the first frame are given, they might vanish for far frames. In such circumstances, RUIC

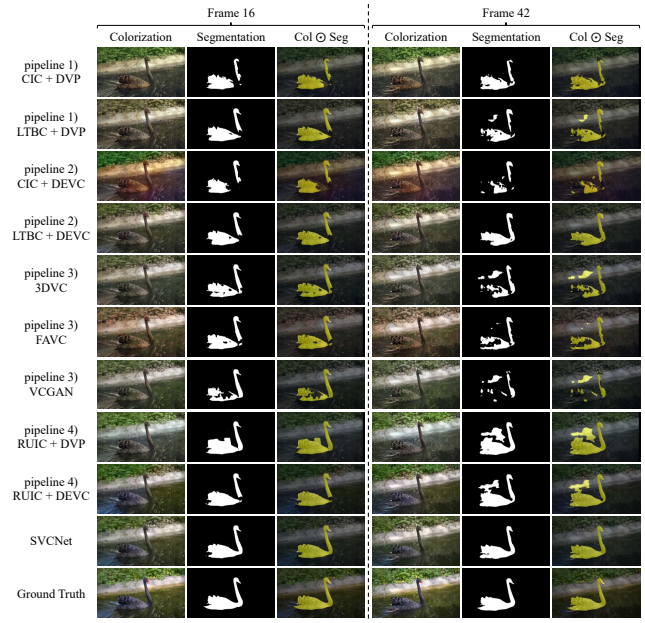


Fig. 7. Illustration of video colorization and segmentation results of SVCNet and other pipelines on the "blackswan" sample of the DAVIS dataset.

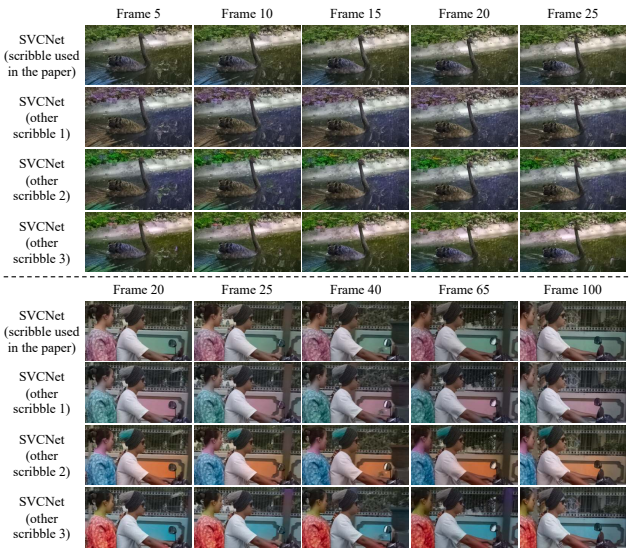


Fig. 8. Illustration of SVCNet results from different color scribbles. The images are extracted from the "blackswan" sample of the DAVIS dataset and the "CoupleRidingMotorbike" sample of the Videvo dataset, respectively. The input color scribbles are omitted.

cannot produce appropriate results, e.g., there are obvious flickering artifacts in RUIC results on the "paragliding-launch" sample. Although DVP and DEVC can bring temporal consistency to the results from RUIC, they are not close enough to the ground truth. We illustrate more examples in Figure 5, where results of RUIC + DVP and RUIC + DEVC are inferior to SVCNet in terms of color vividness and accuracy compared with ground truth.

Colorization Fidelity. According to Table II, SVCNet achieves better performance in terms of both PSNR and SSIM values than other methods. It demonstrates that SVCNet can well use the colors from given color scribbles. Compared with the state-of-the-art method RUIC + DEVC, SVCNet additionally adopts short-range connections in the SSNet; while compared with RUIC + DVP, SVCNet additionally uses the long-range connection. The short-range connections include the previous three and leading three single-frame colorization results and the last output. The long-range connection is the information from the first frame colorization result. Since SVCNet aggregates short-range information, long-range information, and the output of the current time step, it obtains higher colorization fidelity.

Human Preference. We select the top-performed methods FAVC, VCGAN, RUIC, RUIC + BTC, RUIC + DVP, RUIC + DEVC and the SVCNet in the human preference study. The results are shown in Figure 6. It is clear that the proposed SVCNet achieves a better human preference rate than other methods. The experiment demonstrates that colorization results from SVCNet are more temporally consistent and colorful compared with other results.

Color Bleeding Analysis. We further discuss the color bleeding artifacts in this subsection. As shown in Figure 7, the segmentation algorithm produces the clearest segmentation map from the SVCNet results. For instance, the contour of the swan is more continuous for the SVCNet than the other methods. It represents the key objects are more distinguishable from other objects for it. In addition, SVCNet achieves the highest mIoU among all the methods, as concluded in Table II. It demonstrates that the objects of its results are more obvious than the other methods, which denotes that SVCNet has less possibility to encounter color bleeding artifacts. This is achieved by adding an additional segmentation loss at the training, where the weights of the colorization branch are tuned by the segmentation branch.

Colorization Diversity. Since the color scribbles can be diverse, scribble-based video colorization should be a multimodal application. We illustrate some results from different color scribbles in Figure 8. It is clear that SVCNet can generate diverse colors from different input color scribbles.

D. Ablation Study

Experiment Setting. In order to demonstrate the effectiveness of components of the SVCNet, we define seven ablation study settings. The training hyper-parameters are unchanged excluding the ablation study items for each setting. The evaluation is on the original resolution on DAVIS and Videvo datasets, where all the settings share the same color scribbles. The details are concluded as follows:

1) w/o segmentation loss: Drop the segmentation loss L_s^{CP};

2) w/o CPNet pre-training: Drop the CPNet warm-up pre-training stage on video datasets (i.e., only on the ImageNet);

3) w/o short-range connections: Drop the Refinement Module of SSNet with all short-range connections (i.e., y_{t-3}^{ab}, y_{t-2}^{ab}, y_{t-1}^{ab}, y_{t+1}^{ab}, y_{t+2}^{ab}, y_{t+3}^{ab}, and z_{t-1}^{ab});

4) w/o the long-range connection: Drop the Correspondence Module with the long-range connection (i.e., y_1^{ab});

TABLE III
SVCNET ABLATION STUDY ON DAVIS AND VIDEVO DATASETS. THE **RED** COLOR REPRESENTS THE BEST PERFORMANCE.

Ablation study setting	Compared item	Color scribble	DAVIS			Videvo	
			PSNR ↑	SSIM ↑	mIoU ↑	PSNR ↑	SSIM ↑
1) w/o segmentation loss	Training scheme	✓	24.09	0.9378	0.8780	25.04	0.9464
2) w/o CPNet pre-training	Training scheme	✓	19.19	0.8913	0.6464	20.27	0.9067
3) w/o short-range connections	Temporal aggregation	✓	24.60	0.9495	0.8949	25.04	0.9535
4) w/o the long-range connection	Temporal aggregation	✓	24.23	0.9479	0.8919	24.51	0.9511
5) w/o short- and long-range connections	Temporal aggregation	✓	24.01	0.9481	0.8934	24.37	0.9498
6) with 64×128 resolution	Resolution	✓	24.47	0.9522	0.8768	25.26	0.9576
7) with 128×224 resolution	Resolution	✓	24.59	0.9533	0.8960	25.64	0.9589
SVCNet	Full SVCNet	✓	25.71	0.9565	0.9104	26.30	0.9615

Fig. 9. Illustration of different ablation study settings. Samples are selected from the "motocross-jump", "kite-surf", and "bike-packing" samples of the DAVIS dataset, respectively.

5) w/o short- and long-range connections: Drop both short-range connections and the long-range connection;

6) with 64×128 resolution: Change the input image resolution of SVCNet to 64×128. The upsampling ratio of the Super-resolution Module is changed to 8;

7) with 128×224 resolution: Change the input image resolution of SVCNet to 128×224. The upsampling ratio of the Super-resolution Module is changed to 4.

The quantitative results are concluded in Table III and some samples are illustrated in Figure 9.

Training Scheme. For settings 1) and 2), we exclude some parts in the training stages. If dropping the segmentation loss, there are color bleeding artifacts (e.g., some regions of the sky are colorized in green), as shown in Figure 9. If dropping the CPNet warm-up pre-training stage on video datasets, the colorization is wrong. Training all modules without warm-up pre-training is extremely difficult. It is because the CPNet cannot well colorize video frames that have different resolutions with images; meanwhile, the SSNet becomes ineffective when the given colorized frames from the CPNet are not good. In addition, as concluded in Table III, the mIoU of setting 1) and results of all metrics of setting 2) decrease obviously.

Temporal Aggregation. For settings 3-5), we do not use some temporal information. In Figure 9, the colors of frames 22-24 results of setting 3) are not consistent since short-range connections are not used. The colors of far frame results of settings 4) and 5) are also not consistent with the frame 3 result. Since the long-range connection is not used and the color scribbles easily vanish when there are large motions, the colors for far frames are distorted. Also in Table III, excluding temporal information leads to a giant decrease in both PSNR and SSIM values. The experiment results demonstrate that both short-range connections and the long-range connection are significant for the SVCNet.

Resolution. For settings 6) and 7), we change the image resolution. Though reducing the resolution can accelerate the inference speed, the performance of SVCNet drops obviously. Also in most cases, colorization applications do not need a very quick inference time. Considering the balance of colorization quality, inference speed, and memory cost, we use 256×448 as the running resolution.

In conclusion, all the proposed training schemes, temporal aggregation, and network architecture are significant.

TABLE IV
COMPARISON OF THE NUMBER OF PARAMETERS (N_{param}) AND MULTIPLY ACCUMULATES (MACS) ON A PATCH WITH A RESOLUTION OF 256×448 OR 1024×1792 ($MACs_{256}$ AND $MACs_{1024}$) FOR SVCNET.

Module	N_{param}	$MACs_{256}$	$MACs_{1024}$
CPNet	91.475M	251.690G	251.690G
Correspondence Module	26.942M	162.102G	162.102G
Refinement Module	346.608K	28.664G	28.664G
Combination Module	7.084M	20.479G	20.479G
Super-resolution Module	306.432K	33.096G	55.292G
SSNet	34.678M	244.340G	266.537G
SVCNet	126.153M	496.030G	518.226G

E. Computational Costs

The computational costs for the SVCNet are concluded in Table IV. Based on the sparsity of color components, the majority of operations are on a fixed small resolution (i.e., 256×448). Therefore, the computational costs for all modules except for the Super-resolution Module remain the same for different input resolutions. When changing the input resolution

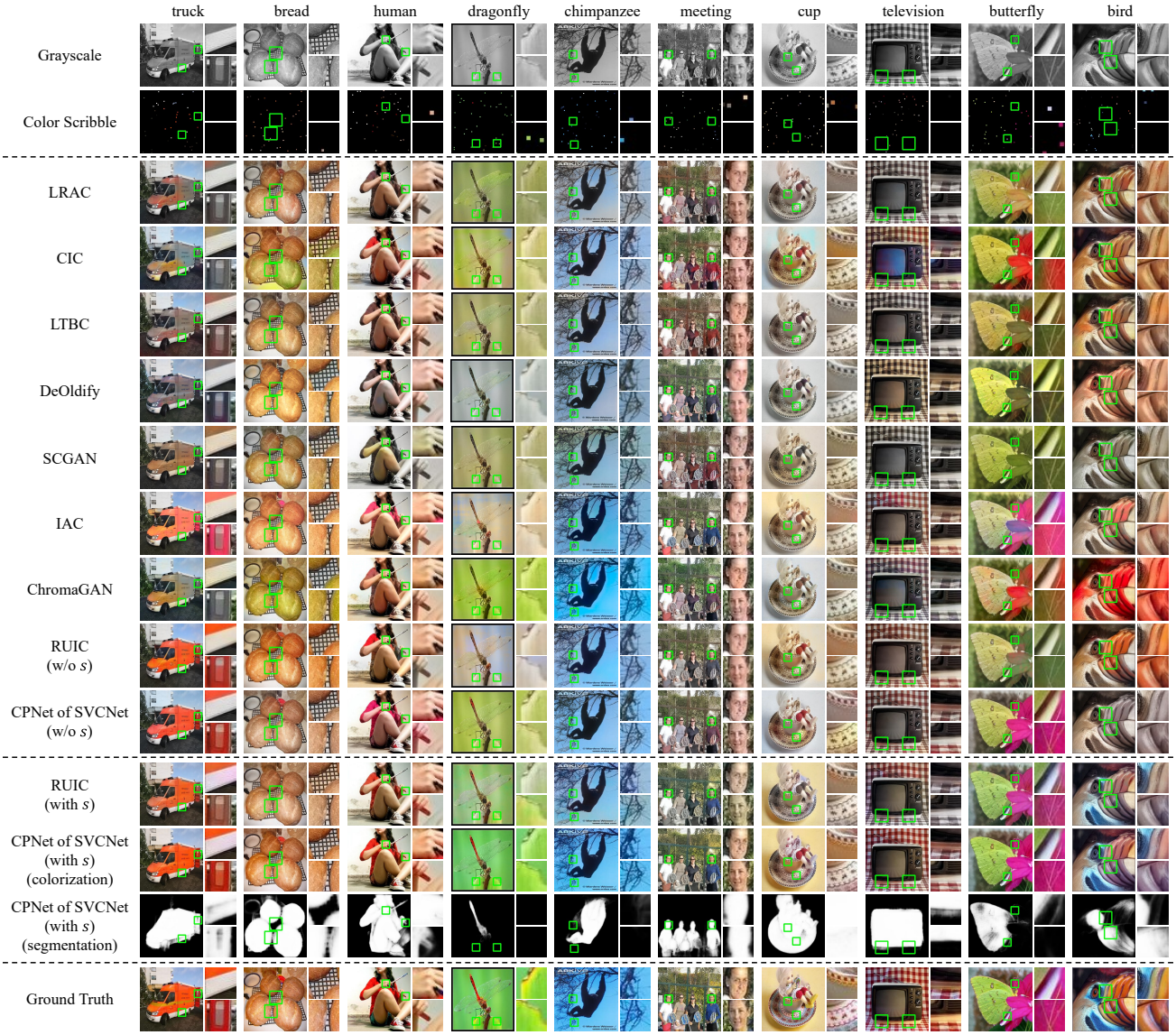


Fig. 10. Illustration of image colorization results of SVCNet and state-of-the-art methods, where RUIC [30] and SVCNet are scribble-based methods. The first and the second rows denote the grayscale and color scribbles, respectively. The last and the second last rows are the ground truth and predicted segmentation maps by SVCNet, respectively. The other rows include the colorization results of different methods. The patches are shown alongside full-resolution images.

to 1024×1792 from 256×448, multiply accumulates (MACs) only increase by 22.197G, which is only 4.5% of MACs of 256×448 resolution (496.030G). This design makes the SVCNet memory-friendly. In addition, using small architectures for the Refinement Module and the Super-resolution Module is enough. The sum of their parameters accounts for less than 1% of the overall parameters of the whole SVCNet. Please refer to Figure 3 (b) and the base channels for them are 16 and 32, respectively. Though small architectures are used, they fulfill the targets well, as shown in Figure 3 (c) and (d), respectively.

F. Image Colorization Experiments

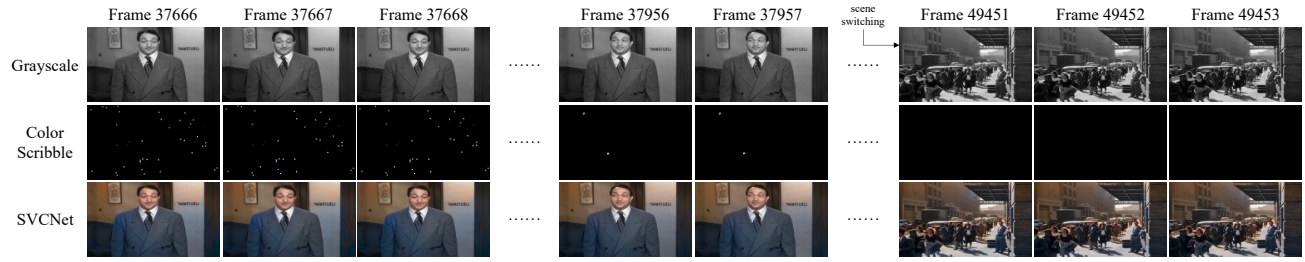
Experiment Setting. In order to further demonstrate the colorization quality of SVCNet, we compare the CPNet of SVCNet with the following baselines:

1) Fully-automatic methods: CIC [52], LTBC [53], LRAC [51], Pix2Pix [54], DeOldify [101], FAVC [9], ChromaGAN [57], SCGAN [59], IAC [58];

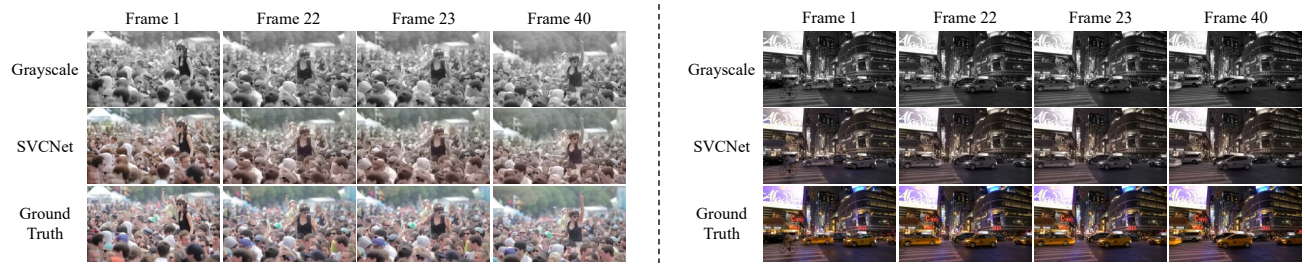
2) Scribble-based method: RUIC [30].

All the methods are trained on ImageNet 1.3 million training set and evaluated on ImageNet 10000 validation set, as defined by [51], [52], [59]. All the methods are trained and evaluated on 256×256 image resolution. We use 40 color scribbles for SVCNet and RUIC in the experiment.

Qualitative Analysis. The qualitative samples are illustrated in Figure 10. Firstly, compared with other methods, there are fewer color bleeding artifacts in the CPNet results (i.e., the color of an object does not permeate through other objects). It is because we use a segmentation branch and a segmentation loss, which helps the network focus on the key objects and separate them from other objects. In the contrast, there are ob-



(a) Failure case: The input video clip is very long and there is scene switching in the full video.



(b) Failure case: There are too many textures in the input video clip. The experiments are conducted on "Festival" and "TimeSquareTraffic" samples from the Videvo dataset.

Fig. 11. Illustration of two common failure cases: (a) The input video clip is too long and there is scene switching; (b) There are too many textures. The frames in sub-figure (a) are from a 1948 grayscale film "The Naked City" with an FPS of 24. The input color scribbles are omitted for sub-figure (b).

TABLE V
COMPARISON OF IMAGE COLORIZATION METHODS AND THE PROPOSED CPNET OF SVCNET ON THE IMAGENET DATASET.

Method	Color scribble	PSNR	SSIM
CIC [52]	/	22.62	0.9153
LTBC [53]	/	24.96	0.9464
LRAC [51]	/	24.49	0.9229
Pix2Pix [54]	/	23.39	0.9386
DeOldify [101]	/	23.14	0.9194
FAVC [9]	/	22.96	0.9146
ChromaGAN [57]	/	23.67	0.9273
SCGAN [59]	/	23.93	0.9470
IAC [58]	/	24.91	0.9110
VCGAN [11]	/	24.58	0.9427
RUIC [30] (w/o s)	/	25.69	0.9526
CPNet of SVCNet (w/o s)	/	25.74	0.9577
RUIC [30] (with s)	✓	28.94	0.9640
CPNet of SVCNet (with s)	✓	31.40	0.9760

vious artifacts for the other methods, e.g., "truck" patches from LRAC, CIC, LTBC, and "chimpanzee", "meeting" patches from RUIC, etc. Secondly, there is fewer color confusion issue (i.e., the colors are semantically wrong for some objects) in the CPNet results even when there are no given color scribbles. For instance, the background of the "dragonfly" sample is colorized to blue for IAC, DeOldify, and RUIC (w/o s). The background of the "cup" sample is colorized not consistently for CIC, IAC, and RUIC (w/o s). The colors of "truck" from LRAC, CIC, LTBC, and ChromaGAN are not consistent. However, the colors of these samples are more reasonable for the CPNet. In conclusion, the proposed CPNet has a stronger ability to perform scribble-based image colorization, which serves as a powerful backbone for the SVCNet.

Quantitative Analysis. The quantitative results are concluded in Table V. On one hand, CPNet obtains better performance than the state-of-the-art scribble-based image colorization method RUIC either without color scribbles (w/o s) or with color scribbles (with s). Especially when using color scribbles, CPNet largely outperforms RUIC, e.g., 2.46 higher PSNR and 0.0120 higher SSIM. Based on the powerful CPNet, SVCNet has the potential to colorize videos with high quality. On the other hand, CPNet outperforms existing methods when using color scribbles. It is because CPNet can well utilize the information from the color scribbles to guide the colorization. Compared with conventional architectures, SVCNet directly uses a pre-trained semantic feature encoder to extract features and has a decoder segmentation branch to predict the segmentation map, which helps the network address the color bleeding artifacts. The network designs also contribute to better performance.

G. Failure Cases and Discussion

The SVCNet can produce high-quality colorful videos by propagating the color scribbles to the grayscale videos. The results are often not plausible when: 1) the video clip is very long and there is scene switching in the full video; 2) there are too many textures, as illustrated in Figure 11.

Firstly, we use a real legacy video for experiments, as shown in Figure 11 (a). When we give the color scribbles for the first frame of a scene (e.g., Frame 37666), the color styles of results for far frames in the scene (e.g., Frames 37666 and 37957) are consistent with the first frame. However, when we colorize the far frames (e.g., Frame 49451, which is approximately 8 minutes later than Frame 37666), the details are not good. The color style of Frame 49451 is also very similar to the first colorized frame (i.e., Frame 37666) because the SVCNet uses only the long-range connection for reference; however, the styles of the two distinct scenes are not always similar. We assume that new color scribbles should be given for far keyframes with a keyframe detection algorithm [102].

Secondly, when there are a lot of details in grayscale videos, the SVCNet is hard to colorize the tiny objects, as shown in Figure 11 (b). In the future, we will develop more powerful color propagation techniques and make SVCNet robust to complicated textures.

V. Conclusion

In this paper, we present the first scribble-based video colorization framework called SVCNet. It includes two sequential sub-networks called color propagation network (CPNet) and spatiotemporal smoothing network (SSNet). The CPNet performs accurate image colorization based on the given color scribbles. Utilizing two feature encoders, it effectively extracts semantics and fuses the information of color scribbles and grayscale images. It also contains two decoder branches, where one for producing color embeddings and the other for generating corresponding segmentation maps. By enforcing the multi-task losses at training, the segmentation branch helps CPNet alleviate color bleeding artifacts. The SSNet post-processes the output from the CPNet, which aggregates short-range connections (neighboring colorized frames), the long-range connection (the first colorized frame), and information of the current time step (the current CPNet output) to achieve good temporal consistency. In addition, we notice that color embeddings are sparse so we set the inference resolution to a fixed small size and we use a Super-resolution Module to recover the larger resolution for HD video applications at the tail of SVCNet. Finally, we compare SVCNet with several state-of-the-art image colorization and video colorization methods. The results demonstrate that SVCNet produces more realistic results and encounters fewer color bleeding artifacts than existing methods.

Acknowledgment

The authors would like to thank the anonymous reviewers and editors for their helpful comments.

References

[1] N. Bonneel, J. Tompkin, K. Sunkavalli, D. Sun, S. Paris, and H. Pfister, "Blind video temporal consistency," *ACM Trans. on Graphics*, vol. 34, no. 6, pp. 1–9, 2015.

[2] W.-S. Lai, J.-B. Huang, O. Wang, E. Shechtman, E. Yumer, and M.-H. Yang, "Learning blind video temporal consistency," in *Proc. ECCV*, 2018, pp. 170–185.

[3] Y. Zhou, X. Xu, F. Shen, L. Gao, H. Lu, and H. T. Shen, "Temporal denoising mask synthesis network for learning blind video temporal consistency," in *Proc. ACMMM*, 2020, pp. 475–483.

[4] C. Lei, Y. Xing, and Q. Chen, "Blind video temporal consistency via deep video prior," in *Proc. NeurIPS*, 2020, pp. 1083–1093.

[5] V. Jampani, R. Gadde, and P. V. Gehler, "Video propagation networks," in *Proc. CVPR*, 2017, pp. 451–461.

[6] S. Meyer, V. Cornillère, A. Djelouah, C. Schroers, and M. Gross, "Deep video color propagation," in *Proc. BMVC*, 2018, pp. 85.1–85.10.

[7] R. Endo, Y. Kawai, and T. Mchizuki, "A practical monochrome video colorization framework for broadcast program production," *IEEE Trans. Broadcast.*, 2020.

[8] R. Wu, H. Lin, X. Qi, and J. Jia, "Memory selection network for video propagation," in *Proc. ECCV*, 2020, pp. 175–190.

[9] C. Lei and Q. Chen, "Fully automatic video colorization with self-regularization and diversity," in *Proc. CVPR*, 2019, pp. 3753–3761.

[10] P. Kouzouglidis, G. Sfikas, and C. Nikou, "Automatic video colorization using 3d conditional generative adversarial networks," in *Proc. ISVC*, 2019, pp. 209–218.

[11] Y. Zhao, L.-M. Po, W.-Y. Yu, Y. A. U. Rehman, M. Liu, Y. Zhang, and W. Ou, "Vcgan: Video colorization with hybrid generative adversarial network," *IEEE Trans. Multimedia*, pp. 1–1, 2022.

[12] B. Zhang, M. He, J. Liao, P. V. Sander, L. Yuan, A. Bermak, and D. Chen, "Deep exemplar-based video colorization," in *Proc. CVPR*, 2019, pp. 8052–8061.

[13] S. Iizuka and E. Simo-Serra, "Deepremaster: temporal source-reference attention networks for comprehensive video enhancement," *ACM Trans. on Graphics*, vol. 38, no. 6, pp. 1–13, 2019.

[14] Z. Wan, B. Zhang, D. Chen, and J. Liao, "Bringing old films back to life," *arXiv preprint arXiv:2203.17276*, 2022.

[15] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015.

[16] C. Liu, J. Yuen, and A. Torralba, "Sift flow: Dense correspondence across scenes and its applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 978–994, 2010.

[17] J. Chen, A. Adams, N. Wadhwa, and S. W. Hasinoff, "Bilateral guided upsampling," *ACM Trans. on Graphics*, vol. 35, no. 6, pp. 1–8, 2016.

[18] S. Guadarrama, R. Dahl, D. Bieber, M. Norouzi, J. Shlens, and K. Murphy, "Pixcolor: Pixel recursive colorization," in *Proc. BMVC*, 2017, pp. 112.1–112.13.

[19] Y. Zhao, Y. Xu, Q. Yan, D. Yang, X. Wang, and L.-M. Po, "D2hnet: Joint denoising and deblurring with hierarchical network for robust night image restoration," in *Proc. ECCV*, 2022.

[20] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung, "A benchmark dataset and evaluation methodology for video object segmentation," in *Proc. CVPR*, 2016, pp. 724–732.

[21] "videvo," https://www.videvo.net.

[22] A. Levin, D. Lischinski, and Y. Weiss, "Colorization using optimization," *ACM Trans. on Graphics*, vol. 23, no. 3, pp. 689–694, 2004.

[23] Y.-C. Huang, Y.-S. Tung, J.-C. Chen, S.-W. Wang, and J.-L. Wu, "An adaptive edge detection based colorization algorithm and its applications," in *Proc. ACMMM*, 2005, pp. 351–354.

[24] L. Yatziv and G. Sapiro, "Fast image and video colorization using chrominance blending," *IEEE Trans. Image Process.*, vol. 15, no. 5, pp. 1120–1129, 2006.

[25] K. Xu, Y. Li, T. Ju, S.-M. Hu, and T.-Q. Liu, "Efficient affinity-based edit propagation using k-d tree," *ACM Trans. on Graphics*, vol. 28, no. 5, pp. 1–6, 2009.

[26] X. Chen, D. Zou, Q. Zhao, and P. Tan, "Manifold preserving edit propagation," *ACM Trans. on Graphics*, vol. 31, no. 6, pp. 1–7, 2012.

[27] L. Xu, Q. Yan, and J. Jia, "A sparse control model for image and video editing," *ACM Trans. on Graphics*, vol. 32, no. 6, pp. 1–10, 2013.

[28] B. Sheng, H. Sun, M. Magnor, and P. Li, "Video colorization using parallel optimization in feature space," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 3, pp. 407–417, 2013.

[29] S. Paul, S. Bhattacharya, and S. Gupta, "Spatiotemporal colorization of video using 3d steerable pyramids," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 8, pp. 1605–1619, 2016.

[30] R. Y. Zhang, J. Y. Zhu, P. Isola, X. Geng, A. S. Lin, T. Yu, and A. A. Efros, "Real-time user-guided image colorization with learned deep priors," *ACM Trans. on Graphics*, vol. 36, no. 4, p. 119, 2017.

[31] L. Zhang, C. Li, T.-T. Wong, Y. Ji, and C. Liu, "Two-stage sketch colorization," *ACM Trans. on Graphics*, vol. 37, no. 6, pp. 1–14, 2018.

[32] Y. Ci, X. Ma, Z. Wang, H. Li, and Z. Luo, "User-guided deep anime line art colorization with conditional adversarial networks," in *Proc. ACMMM*, 2018, pp. 1536–1544.

[33] Y. Zhao, L.-M. Po, T. Lin, X. Wang, K. Liu, Y. Zhang, W.-Y. Yu, P. Xian, and J. Xiong, "Legacy photo editing with learned noise prior," in *Proc. WACV*, 2021, pp. 2103–2112.

[34] E. Reinhard, M. Adhikhmin, B. Gooch, and P. Shirley, "Color transfer between images," *IEEE Comput. Graph. Appl.*, vol. 21, no. 5, pp. 34–41, 2001.

[35] T. Welsh, M. Ashikhmin, and K. Mueller, "Transferring color to greyscale images," in *Proc. ACM SIGGRAPH*, 2002, pp. 277–280.

[36] R. Ironi, D. Cohen-Or, and D. Lischinski, "Colorization by example," in *Proc. EGSR*, 2005, pp. 201–210.

[37] Y.-W. Tai, J. Jia, and C.-K. Tang, "Local color transfer via probabilistic segmentation by expectation-maximization," in *Proc. CVPR*, vol. 1, 2005, pp. 747–754.

[38] G. Charpiat, M. Hofmann, and B. Schölkopf, "Automatic image colorization via multimodal predictions," in *Proc. ECCV*, 2008, pp. 126–139.

[39] A. Y.-S. Chia, S. Zhuo, R. K. Gupta, Y.-W. Tai, S.-Y. Cho, P. Tan, and S. Lin, "Semantic colorization with internet images," *ACM Trans. on Graphics*, vol. 30, no. 6, pp. 1–8, 2011.

[40] A. Bugeau, V.-T. Ta, and N. Papadakis, "Variational exemplar-based image colorization," *IEEE Trans. Image Process.*, vol. 23, no. 1, pp. 298–307, 2013.

[41] M. He, D. Chen, J. Liao, P. V. Sander, and L. Yuan, "Deep exemplar-based colorization," *ACM Trans. on Graphics*, vol. 37, no. 4, p. 47, 2018.

[42] T.-H. Sun, C.-H. Lai, S.-K. Wong, and Y.-S. Wang, "Adversarial colorization of icons based on structure and color conditions," *Proc. ACMMM*, pp. 683–691, 2019.

[43] B. Li, Y.-K. Lai, M. John, and P. L. Rosin, "Automatic example-based image colorization using location-aware cross-scale matching," *IEEE Trans. Image Process.*, vol. 28, no. 9, pp. 4606–4619, 2019.

[44] Z. Xu, T. Wang, F. Fang, Y. Sheng, and G. Zhang, "Stylization-based architecture for fast deep exemplar colorization," in *Proc. CVPR*, 2020, pp. 9363–9372.

[45] Y. Bai, C. Dong, Z. Chai, A. Wang, Z. Xu, and C. Yuan, "Semantic-sparse colorization network for deep exemplar-based colorization," in *Proc. ECCV*, 2022, pp. 505–521.

[46] Z. Sheng, H.-L. Shen, B. Yao, and H. Zhang, "Guided colorization using mono-color image pairs," *IEEE Trans. Image Process.*, vol. 32, pp. 905–920, 2023.

[47] H. Bahng, S. Yoo, W. Cho, D. Keetae Park, Z. Wu, X. Ma, and J. Choo, "Coloring with words: Guiding image colorization through text-based palette generation," in *Proc. ECCV*, 2018, pp. 431–447.

[48] C. Zou, H. Mo, C. Gao, R. Du, and H. Fu, "Language-based colorization of scene sketches," *ACM Trans. on Graphics*, vol. 38, no. 6, pp. 1–16, 2019.

[49] Z. Chang, S. Weng, Y. Li, S. Li, and B. Shi, "L-coder: Language-based colorization with color-object decoupling transformer," in *Proc. ECCV*, 2022, pp. 360–375.

[50] Z. Cheng, Q. Yang, and B. Sheng, "Deep colorization," in *Proc. ICCV*, 2015, pp. 415–423.

[51] G. Larsson, M. Maire, and G. Shakhnarovich, "Learning representations for automatic colorization," in *Proc. ECCV*, 2016, pp. 577–593.

[52] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," in *Proc. ECCV*, 2016, pp. 649–666.

[53] S. Iizuka, E. Simo-Serra, and H. Ishikawa, "Let there be color!: joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification," *ACM Trans. on Graphics*, vol. 35, no. 4, p. 110, 2016.

[54] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. CVPR*, 2017, pp. 1125–1134.

[55] A. Royer, A. Kolesnikov, and C. H. Lampert, "Probabilistic image colorization," in *Proc. BMVC*, 2017, pp. 85.1–85.12.

[56] J. Zhao, J. Han, L. Shao, and C. G. Snoek, "Pixelated semantic colorization," *Int. J. Comput. Vis.*, vol. 128, no. 4, pp. 818–834, 2020.

[57] P. Vitoria, L. Raad, and C. Ballester, "Chromagan: Adversarial picture colorization with semantic class distribution," in *Proc. WACV*, 2020, pp. 2445–2454.

[58] J.-W. Su, H.-K. Chu, and J.-B. Huang, "Instance-aware image colorization," in *Proc. CVPR*, 2020, pp. 7968–7977.

[59] Y. Zhao, L.-M. Po, K.-W. Cheung, W.-Y. Yu, and Y. A. U. Rehman, "Scgan: Saliency map-guided colorization with generative adversarial network," *IEEE Trans. Circuits Syst. Video Technol.*, pp. 1–1, 2020.

[60] M. Kumar, D. Weissenborn, and N. Kalchbrenner, "Colorization transformer," in *Proc. ICLR*, 2021. [Online]. Available: https://openreview.net/forum?id=5NA1PinlGFu

[61] Y. Wu, X. Wang, Y. Li, H. Zhang, X. Zhao, and Y. Shan, "Towards vivid and diverse image colorization with generative color prior," in *Proc. ICCV*, 2021, pp. 14 377–14 386.

[62] R. Pucci, C. Micheloni, and N. Martinel, "Collaborative image and object level features for image colourisation," in *Proc. CVPRW*, 2021, pp. 2160–2169.

[63] G. Kong, H. Tian, X. Duan, and H. Long, "Adversarial edge-aware image colorization with semantic segmentation," *IEEE Access*, vol. 9, pp. 28 194–28 203, 2021.

[64] C. Saharia, W. Chan, H. Chang, C. Lee, J. Ho, T. Salimans, D. Fleet, and M. Norouzi, "Palette: Image-to-image diffusion models," in *Proc. ACM SIGGRAPH*, 2022, pp. 1–10.

[65] G. Kim, K. Kang, S. Kim, H. Lee, S. Kim, J. Kim, S.-H. Baek, and S. Cho, "Bigcolor: Colorization using a generative color prior for natural images," in *Proc. ECCV*, 2022, pp. 350–366.

[66] Z. Huang, N. Zhao, and J. Liao, "Unicolor: A unified framework for multi-modal colorization with transformer," *ACM Trans. on Graphics*, vol. 41, no. 6, pp. 1–16, 2022.

[67] C. Lei, Y. Xing, H. Ouyang, and Q. Chen, "Deep video prior for video consistency and propagation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 1, pp. 356–371, 2022.

[68] C. Lei, X. Ren, Z. Zhang, and Q. Chen, "Blind video deflickering by neural filtering with a flawed atlas," *arXiv preprint arXiv:2303.08120*, 2023.

[69] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. Torr, and S.-M. Hu, "Global contrast based salient region detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 569–582, 2014.

[70] C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang, "Saliency detection via graph-based manifold ranking," in *Proc. CVPR*, 2013, pp. 3166–3173.

[71] G. Li and Y. Yu, "Visual saliency based on multiscale deep features," in *Proc. CVPR*, 2015, pp. 5455–5463.

[72] L. Wang, L. Wang, H. Lu, P. Zhang, and X. Ruan, "Saliency detection with recurrent fully convolutional networks," in *Proc. ECCV*, 2016, pp. 825–841.

[73] N. Liu and J. Han, "Dhsnet: Deep hierarchical saliency network for salient object detection," in *Proc. CVPR*, 2016, pp. 678–686.

[74] X. Zhang, T. Wang, J. Qi, H. Lu, and G. Wang, "Progressive attention guided recurrent network for salient object detection," in *Proc. CVPR*, 2018, pp. 714–722.

[75] J.-J. Liu, Q. Hou, M.-M. Cheng, J. Feng, and J. Jiang, "A simple pooling-based design for real-time salient object detection," in *Proc. CVPR*, 2019, pp. 3917–3926.

[76] Y.-H. Wu, Y. Liu, L. Zhang, M.-M. Cheng, and B. Ren, "Edn: Salient object detection via extremely-downsampled network," *IEEE Trans. Image Process.*, 2022.

[77] L. Wang, H. Lu, X. Ruan, and M.-H. Yang, "Deep networks for saliency detection via local estimation and global search," in *Proc. CVPR*, 2015, pp. 3183–3192.

[78] T. Zhao and X. Wu, "Pyramid feature attention network for saliency detection," in *Proc. CVPR*, 2019, pp. 3085–3094.

[79] L. Zhang, J. Wu, T. Wang, A. Borji, G. Wei, and H. Lu, "A multistage refinement network for salient object detection," *IEEE Trans. Image Process.*, vol. 29, pp. 3534–3545, 2020.

[80] A. Siris, J. Jiao, G. K. Tam, X. Xie, and R. W. Lau, "Scene context-aware salient object detection," in *Proc. ICCV*, 2021, pp. 4156–4166.

[81] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. CVPR*, 2015, pp. 3431–3440.

[82] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. CVPR*, 2017, pp. 2881–2890.

[83] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, 2017.

[84] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proc. ECCV*, 2018, pp. 801–818.

[85] M. Yang, K. Yu, C. Zhang, Z. Li, and K. Yang, "Denseaspp for semantic segmentation in street scenes," in *Proc. CVPR*, 2018, pp. 3684–3692.

[86] K. Liu, "Rm3d: Robust data-efficient 3d scene parsing via traditional and learnt 3d descriptors-based semantic region merging," *Int. J. Comput. Vis.*, pp. 1–30, 2022.

[87] K. Liu, A. Xiao, X. Zhang, S. Lu, and L. Shao, "Fac: 3d representation learning via foreground aware feature contrast," *arXiv preprint arXiv:2303.06388*, 2023.

[88] K. Liu, Y. Zhao, Q. Nie, Z. Gao, and B. M. Chen, "Weakly supervised 3d scene segmentation with region-level boundary awareness and instance discrimination," in *Proc. ECCV*, 2022, pp. 37–55.

[89] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. MICCAI*, 2015, pp. 234–241.

[90] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume," in *Proc. CVPR*, 2018, pp. 8934–8943.

[91] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. ICLR*, 2015. [Online]. Available: https://arxiv.org/abs/1409.1556

[92] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. AISTATS*, 2010, pp. 249–256.

[93] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2015. [Online]. Available: https://openreview.net/forum?id=8gmWwjFyLj
[94] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, 2016, pp. 770–778.
[95] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. Change Loy, "Esrgan: Enhanced super-resolution generative adversarial networks," in *Proc. ECCVW*, 2018, pp. 0–0.
[96] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. ICML*, vol. 30, no. 1, 2013, p. 3.
[97] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," *arXiv preprint arXiv:1607.08022*, 2016.
[98] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli *et al.*, "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, 2004.
[99] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep high-resolution representation learning for human pose estimation," in *Proc. CVPR*, 2019.
[100] Y. Yuan, X. Chen, X. Chen, and J. Wang, "Segmentation transformer: Object-contextual representations for semantic segmentation," *arXiv preprint arXiv:1909.11065*, 2019.
[101] "Deoldify," https://github.com/jantic/DeOldify.
[102] X. Yan, S. Z. Gilani, H. Qin, M. Feng, L. Zhang, and A. Mian, "Deep keyframe detection in human action videos," *arXiv preprint arXiv:1804.10021*, 2018.