VCGAN: Video Colorization with Hybrid Generative Adversarial Network

Yuzhi Zhao, *Graduate Student Member, IEEE,* Lai-Man Po, *Senior Member, IEEE,* Wing-Yin Yu, *Graduate Student Member, IEEE,* Yasar Abbas Ur Rehman, *Member, IEEE,* Mengyang Liu, Yujia Zhang, Weifeng Ou

***Abstract*—We propose a hybrid recurrent Video Colorization with Hybrid Generative Adversarial Network (VCGAN), an improved approach to video colorization using end-to-end learning. The VCGAN addresses two prevalent issues in the video colorization domain: Temporal consistency and unification of colorization network and refinement network into a single architecture. To enhance colorization quality and spatiotemporal consistency, the mainstream of generator in VCGAN is assisted by two additional networks, i.e., global feature extractor and placeholder feature extractor, respectively. The global feature extractor encodes the global semantics of grayscale input to enhance colorization quality, whereas the placeholder feature extractor acts as a feedback connection to encode the semantics of the previous colorized frame in order to maintain spatiotemporal consistency. If changing the input for placeholder feature extractor as grayscale input, the hybrid VCGAN also has the potential to perform image colorization. To improve the consistency of far frames, we propose a dense long-term loss that smooths the temporal disparity of every two remote frames. Trained with colorization and temporal losses jointly, VCGAN strikes a good balance between color vividness and video continuity. Experimental results demonstrate that VCGAN produces higher-quality and temporally more consistent colorful videos than existing approaches.**

***Index Terms*—Colorization, Generative Adversarial Networks, Placeholder Feature Extractor.**

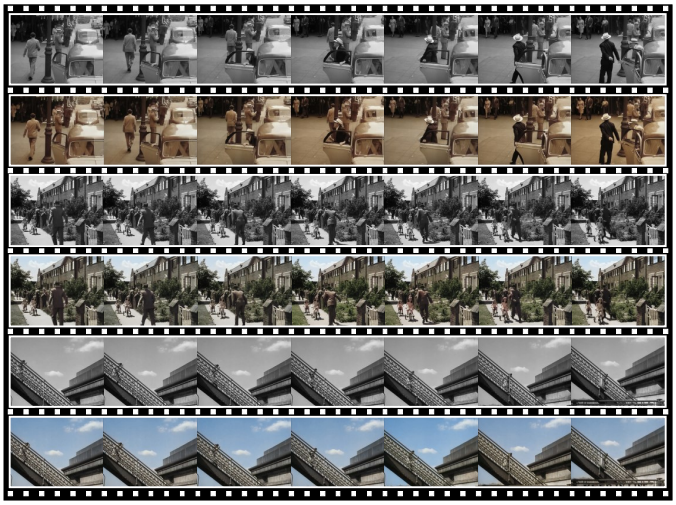


Fig. 1. Colorization results of a 1948 American grayscale film 'The Naked City' by the proposed VCGAN. Different rows represent different scenes in the film. The interval of frames equals to 5. Please see **https://github.com/zhaoyuzhi/VCGAN** for supplementary materials.

I. Introduction

THERE are many legacy movies and historical videos in black-and-white format. Restricted by the photography technology at that time, it was extremely hard to preserve color information. If the grayscale videos are painted with reasonable colors, they could show the vividness of the past time. Recently, the convolutional neural networks (CNNs) automate the process of grayscale image colorization [1]–[3], [3]–[19]. To predict plausible colorized image, researchers combined many objective functions such as L1 loss, MSE loss, perceptual loss [20], KL loss [6], and classification loss on each pixel [4] or advanced training schemes like adversarial training [21] and coarse-to-fine scheme [22]. However, those image colorization algorithms cannot be directly utilized to colorize grayscale videos, since they are unable to learn spatiotemporal consistency. Since adjacent frames in a video are temporally correlated, the additional spatiotemporal constraints are significant for video colorization applications.

Video colorization methods can be categorized into three classes: exemplar-guided [17], [18], [23]–[26], task-independent [27], [28], and fully-automatic [14]–[16]. On the one hand, the earlier video colorization methods based on manually defining color scribbles on grayscale frames [23]–[25]. However, the results are not stable when the video scene changes. To address the issue, exemplar image-based methods [18], [26] matched the similarity between input frames and reference image. Similarly, Jampani *et al.* [17] used several colorized frames as exemplars for the following frames. However, the results from these approaches are still sensitive to scene disparity. On the other hand, to alleviate the effort of selecting proper examples, task-independent models such as [27], [28] aimed to post-process framewise colorization results. By adding temporal coherence to independent colorized frames, they may fit video colorization. For instance, Lai et al. [28] proposed a temporal smoothing network and used an optical flow estimator [29]. It minimizes the color difference between every two neighbouring individual colorized frames.

A preprint version of the manuscript under revision of IEEE Transactions on Multimedia. *(Corresponding author: Yuzhi Zhao.)*

Y. Zhao, L.-M. Po, W.-Y. Yu, Y. Zhang and W. Ou are with the Department of Electronic Engineering, City University of Hong Kong, Hong Kong (e-mail: yzzhao2-c@my.cityu.edu.hk; eelmpo@cityu.edu.hk; wingyinyu8-c@my.cityu.edu.hk; yzhang2383-c@my.cityu.edu.hk; weifengou2-c@my.cityu.edu.hk).

Y.-A.-U. Rehman is with TCL Corporate Research Hong Kong, Hong Kong (e-mail: yasar.abbas@my.cityu.edu.hk).

M. Liu is with Tencent Video, Tencent Holdings Ltd, China (e-mail: mengyaliu7-c@my.cityu.edu.hk).

However, the results are highly based on image colorization algorithms, thus the results are not continuous enough.

Based on the concept from the task-independent algorithms, fully-automatic methods [14]–[16] further enhance the spatiotemporal consistency of output. Recently, Kouzouglidis et al. [15] used 3D-CNN that incorporates spatiotemporal relations explicitly. However, for a long sequence, each segment is still independently colorized due to the memory limit. To reduce memory consumption and process longer video, Lei et al. [14] separated the video colorization into a colorization step for single frame rendering and a smoothing step for temporal refinement. They adopted a hypercolumn [2] to augment the input. However, it still needs large memory consumption and misses the first frame at inference.

In order to address the issues, we propose to combine both image and video colorization into a hybrid architecture by an end-to-end video colorization generative adversarial network (VCGAN). We claim two main benefits of the hybrid model: 1) One model can be applied to both image and video colorization; 2) Providing reference frame for initialization so that no frame is lost. Firstly, the single image is viewed as a "static" video that VCGAN unifies the two tasks with the same architecture. In order to process videos with arbitrary length, we assume that the continuous frames satisfy the Markov chain and its transition function is implied in the model. Therefore, VCGAN is constructed as a recurrent architecture processing input frames sequentially during forward propagation. Secondly, since the VCGAN has the ability to process a single image (i.e., the first frame of video), there are no frames lost.

Regarding the VCGAN architecture, the generator includes a mainstream encoder-decoder and two feature extractors. The mainstream adopts the U-Net structure [30], which ensures the network preserves low-level details and edges. The weights of two feature extractors are loaded from ResNet-50 [31]. The first global feature extractor extracts the semantics of the input grayscale frame of the current time, which provides high-level information for the network to better the colors for objects [2], [4], [5], [32]. The second placeholder feature extractor receives the last colorized frame, in order to enhance the spatiotemporal continuity by the recurrent connection. The output features of both extractors are concatenated to the encoder of the mainstream. In addition, we adopt a patch-based structure for the VCGAN discriminator.

Regarding the training, there are two stages for VCGAN including single frame and video colorization, respectively. Since the total numbers and diversity of frames in the video datasets are much less than image datasets, we firstly use a large-scale image dataset ImageNet [33] to train VCGAN. The first stage provides good initialization weights, which ensures VCGAN has plausible image colorization quality. Then at the second stage, it is optimized with both colorization and spatiotemporal smoothing objectives, using video datasets such as DAVIS [34] and Videvo [35]. In addition, we improve the temporal smoothness of colorized frames by enforcing an additional dense long-term loss at the second stage. It models the dense connections of each remote frame, which is beneficial for VCGAN to maintain color continuity for distant frames. We evaluate the proposed VCGAN in terms of both image and video colorization quality on the benchmark datasets. Experimental results demonstrate that VCGAN can produce high-quality colorizations than the well-known methods. Some results produced by VCGAN are shown in Figure 1.

In general, there are three main contributions of this paper:

1) A hybrid recurrent VCGAN framework is proposed to integrate both image and video colorization applications by two training stages;

2) A dense long-term loss is proposed that there are fundamentally no flicking artifacts of generated frames;

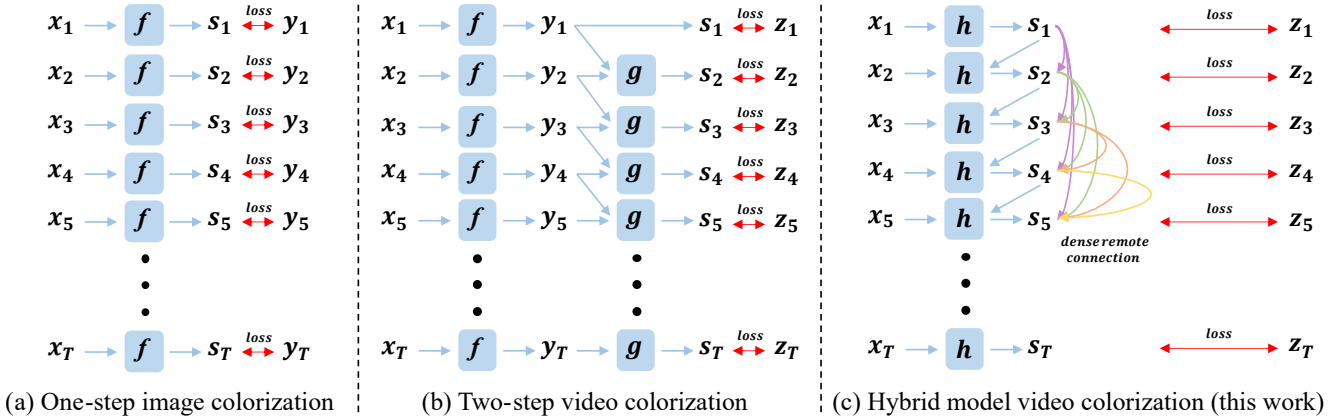
3) Comprehensive experiments are conducted to assess the VCGAN architecture on both single image and video colorization applications. The VCGAN achieves state-of-the-art performances on benchmark datasets compared with some well-known algorithms.

II. Related Work

Image Colorization. There were two categories of image colorization methods: exemplar-based and fully-automatic. The exemplar-based methods are based on additional user-given information such as color scribbles [23]–[25], [32] and example colorful images [26], [36]–[38]. For instance, Levin *et al.* [23] assumed adjacent pixels with the same illuminances should have similar colors and developed an optimization-based system based on the assumption. Welsh *et al.* [36] attached colors from example images to grayscale input by matching spatial features of them. However, these algorithms require accurate hints (e.g., color pixels or similar RGB images) for producing high-quality colorizations, which is labor-intensive.

To alleviate the effort of selecting proper references, fully-automatic image colorization methods [1]–[13] directly learn the mapping from grayscale images to their color embeddings based on deep learning. Cheng *et al.* [1] firstly utilized a deep neural network to colorize images based on three levels of features. However, the performance is limited due to hand-crafted features and a tiny network structure. To improve generation quality, researchers used semantics extracted by pre-trained VGG-Net [39] or ResNet [31]. For instance, Larsson *et al.* [2] adopted a VGG-Net-based hyper-column to extract multi-level representation of grayscale. Iizuka *et al.* [5] used two-stream networks for extracting both low-level and high-level information. While Zhang *et al.* [4] directly adopted VGG-16 as backbone with a color classification loss and category-balancing technique. To augment the colorization for significant objects in an image, Zhao *et al.* [12] used saliency map to aid the learning of colorization and Su *et al.* [13] includes instance segmentation in colorization system.

Video Colorization. There are three classes of video colorization algorithms: exemplar-guided [17]–[19], [23]–[26], task-independent [27], [28] and fully-automatic [14]–[16]. The earlier works were mainly exemplar-guided including propagating the user scribbles [23]–[25], attaching the colors from colorized frames [17] or given images [26] to the rest of frames. Recently, CNNs improve colorization quality since it effectively extracts features from the input [2], [4], [5]. For instance, Zhang *et al.* [18] matched the features between



(a) One-step image colorization (b) Two-step video colorization (c) Hybrid model video colorization (this work)

Fig. 2. Illustration of different connection types of (a) One-step image colorization [1]–[5], (b) Two-step video colorization [14], and (c) the proposed hybrid VCGAN, where x is the input, and s is the video colorization result. f, g, and h represent CNNs. The color lines in (c) indicate that the dense remote connections of generated frames modeled by VCGAN. The losses are computed between s and ground truth y (image colorization) or z (video colorization).

the reference image and input frames to guide colorization. Jampani *et al.* [17] used few colorized frames as references and then propagated them to the whole video. However, their results are plausible when the scene disparity of examples and grayscale frames can be ignored.

Many image colorization algorithms obtain good colorization quality; however, directing using them to each video frame independently often leads to temporal inconsistencies. Thus, the task-independent methods were proposed to explicitly encode the temporal consistency of the independently colorized frames. Bonneel *et al.* [27] addressed the issue by minimizing the disparity of warped frame and next frame with least-square energy. Lai *et al.* [28] introduced a transformation network that post-processes the frames, with an optical flow guidance. The network was trained by both temporal and perceptual loss [20] to strike a balance between temporal coherence and spatial quality. However, the refined frames are still not continuous enough, since the image colorization and temporal refinement networks are not trained collaboratively. To further automate the video colorization pipeline, Lei *et al.* [14] proposed a multimodal automatic system that produced four possible colorized videos. To enhance the color consistency, they performed the K-nearest neighbor (KNN) search that builds a connection between color and spatial location. However, the generated images are not colorful enough.

Generative Adversarial Network for Colorization. GAN was first proposed by Goodfellow *et al.* [21], including two neural networks (i.e., generator and discriminator) that compete against each other. For colorization, GAN was used to enhance the vividness of colorized images [3] or produce diverse results [8], [40]. Isola *et al.* [3] proposed a general Pix2Pix framework for paired images transformation. Experimental analysis proved that adversarial training strategy helps in preserving details and enhancing the perceptual quality. It was enhanced by pix2pixHD framework [22] for high-resolution images. To obtain diverse colorization, Cao *et al.* [8] directly added noise to the first three layers of encoder while Zhu *et al.* [40] introduced a cLR-GAN model including variational training to strengthen the output diversity.

III. Methodology

A. Problem Formulation

Given a grayscale input video, the output colorized video should satisfy two conditions. Firstly, the color of generated frames should be similar to ground truth. Secondly, the temporal disparity of adjacent frames in the colorized video should be small, i.e., there is almost no flickering effect in the colorized video. Both of the conditions are equally crucial for video colorization.

Suppose the frames of input grayscale video with length T is represented as a sequence $X = \{x_1, x_2, ..., x_T\}$. The corresponding results processed by image colorization algorithms can be represented as $Y = \{y_1, y_2, ..., y_T\}$ and the ground truth colorful video frames are $Z = \{z_1, z_2, ..., z_T\}$. Note that, the framewise color similarity of Y should be highly comparable with Z. However, the frames of Y are temporally discontinuous, since the single image colorization methods [1]–[5] only learn one-step conditional distribution $p(Y|X)$. To address the issue of discontinuity, current video colorization methods [14], [18], [28] finetune the results from Y by another refinement network. They learn a two-step joint distribution $p(Z|Y)p(Y|X)$. Under these conditions, the mapping function can be factorized as:

$$p(Z|X,Y) = \prod_{t=2}^{T} p(z_t|y_t, y_{t-1}) p(y_t|x_t) p(y_{t-1}|x_{t-1}). \quad (1)$$

Specifically, the generated frame contains the information of previous the frame x_{t-1} and the current frame x_t; however, there is no direct connection between them. Normally, the $p(y_t|x_t)$ is implemented by an image colorization network, which is trained to generate inconsistent Y by individually colorizing grayscale frames. Then, a refinement network is used to post-process continuous two frames, i.e., $p(z_t|y_t, y_{t-1})$. It is difficult to control the video consistency of generated frames if the networks are trained individually [28]. Although adopting a joint training scheme [14], the system is too large thus the optimization is difficult to perform. To address this problem, we alternatively learn the direct mapping from X

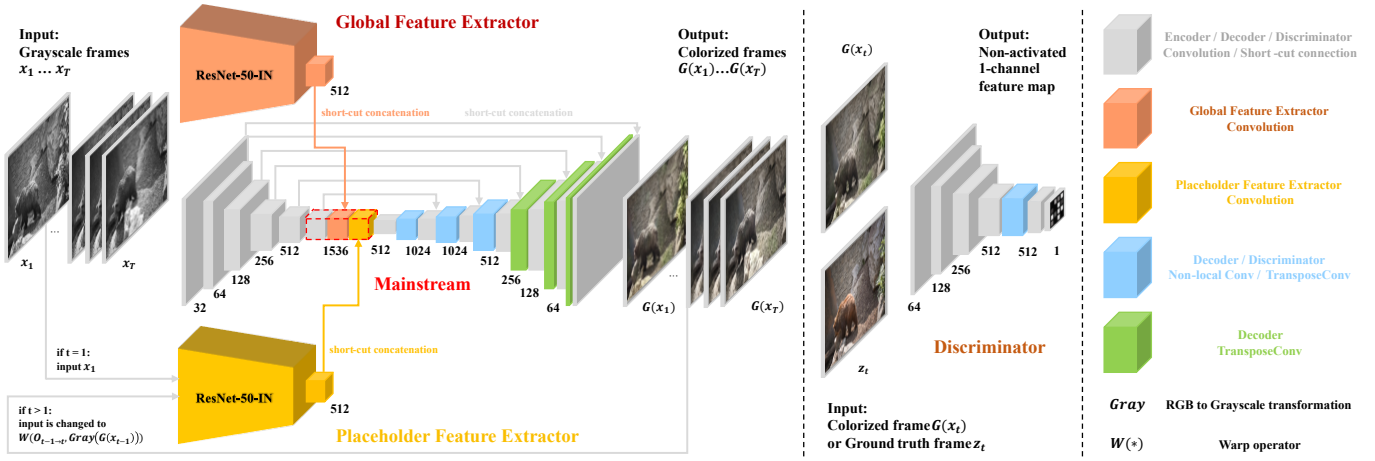


Fig. 3. Illustration of the architecture of the proposed VCGAN. It consists of the generator (left part) and the discriminator (middle part). The right part annotates the different blocks. The generator receives a grayscale frame input at the first frame position. For the following frames, the generator receives a grayscale frame of the current time and last colorized frame as input. The discriminator receives the colorized frame or ground truth frame.

to Z. Therefore, the inference procedure of the proposed VCGAN is represented as:

$$p(Z|X) = \prod_{t=2}^{T} p(z_t|x_t, z_{t-1}) p(z_1|x_1). \tag{2}$$

The two conditional distributions $p(z_t|x_t, z_{t-1})$ and $p(z_1|x_1)$ are combined into one model by a placeholder feature extractor. In addition, the proposed VCGAN is recurrent since the previous colorized frame becomes the input for the colorization process of the next frame, which is optimized to be close to ground truth z_{t-1}. Thus, VCGAN is a hybrid end-to-end model that colorizes grayscale frames sequentially. Since there is no previous frame as a guidance, VCGAN generates the first frame z_1 only based on the initial input frame x_1. For such case, it is viewed as a special video colorization issue, i.e., only one frame in the video. For the following frames, VCGAN does not produce intermediate variables and it is optimized by colorization and temporal smoothing objectives jointly. The recurrent architecture explicitly enforces VCGAN to synthesize more temporally consistent results. Figure 2 illustrates the different connection types of the two aforementioned representative video colorization approaches and VCGAN.

B. VCGAN Architecture

The hierarchical VCGAN consists of four main parts: global feature extractor, placeholder feature extractor, mainstream encoder-decoder, and discriminator, as shown in Figure 3. The first three components constitute the generator. The mainstream adopts U-Net structure [30] that executes skip connection between each encoder layer i and decoder layer $n-i$ with the same spatial resolution, where n is the total number of mainstream layers. It promotes the decoder to preserve low-level details and facilitates the convergence of the whole system since the gradients easily pass to encoder layers. The non-local blocks [41] are attached to bottom layers of the decoder, which strengthen the details using cues from spatially related pixels.

The global feature extractor and placeholder feature extractor utilize a fully convolutional ResNet-50 network [31] architecture, both of which are pre-trained on ImageNet [33]. Since the colorization highly depends on global information [2], [4], [5], the global feature extractor distills semantics from input effectively. While the placeholder feature extractor reserves the information of the last frame to enhance temporal consistency. The outputs of two extractors are concatenated to mainstream encoder for feature fusion. We adopt the PatchGAN discriminator [3] to produce a 1-channel matrix corresponding to input resolutions. It contains fewer parameters than the original 1×1 PixelGAN yet enhances the perceptual quality of generated samples. It also encourages sharper edges and colors.

C. Two-stage Training Schedule

In order to ensure that VCGAN produces perceptually plausible colorizations, the training process is divided into two stages. At the first training stage, the VCGAN performs single image colorization. The large ImageNet dataset [33] is utilized for training since it contains much more diverse modes and categories compared with common video datasets [34], [35]. After its convergence, VCGAN is eligible to produce a single colorful image with high pixel accuracy.

At the second training stage, VCGAN is trained as a Markov Chain that performs a sliding window scheme to select continuous frames. For the first frame, the inputs for two feature extractors and mainstream are the same, i.e., VCGAN learns $p(z_1|x_1)$ (see equation (2)). For the following frames (e.g., time n), the output of time $n-1$ is first converted to grayscale and warped using forward flow from time $n-1$ to n. Then, the warped image replaces the grayscale input for the placeholder feature extractor. The processes can be defined as:

$$s_t = \begin{cases} G(x_1), & t = 1, \\ G(x_t, i_t), & t > 1. \end{cases} \tag{3}$$

$$i_t = W(O_{t-1\to t}, Gray(p_{t-1})), \tag{4}$$

where s_t and i_t are the output of VCGAN generator and input for placeholder feature extractor when $t > 1$. The network $G(*)$ represents the VCGAN generator. The operator $W(*)$ warps input frame under the guidance of given optical flow $O_{t-1\to t}$, and the operator $Gray(*)$ converts RGB images to grayscale by a linear transformation. Note that the operations in the warped image $W(*)$ and $Gray(*)$ are fixed; therefore i_t is proportional to s_{t-1}. Therefore, VCGAN utilizes the information from last output at the second training stage, which satisfies the representation of equation (2).

This design unifies both image and video colorization. Compared with single image colorization algorithms [2], [4], [5], the placeholder feature extractor reserves a place for recurrent feedback. Moreover, it encourages VCGAN to minimize the color discrepancy between neighbouring frames.

D. Objectives

Directly optimizing a conditional GAN framework with adversarial loss often leads to failure. Thus, we pre-train the generator to produce relatively plausible results to stabilize GAN training based on the following objective:

$$L_{1st} = \lambda_1 L_1 + \lambda_p L_p, \tag{5}$$

where L_1 and L_p denote pixel-level reconstruction loss and perceptual loss [20], respectively. The losses are defined as:

$$L_1 = \mathbb{E}[||s_t - z||_1], \tag{6}$$

$$L_p = \mathbb{E}[||\phi_l(s_t) - \phi_l(z)||_1], \tag{7}$$

where s_t and z represent the colorized image (see equation (3)) and corresponding ground truth, respectively. At the first stage, $t = 1$. The function $\phi_l(*)$ outputs the features from the l-th layer of a pre-trained network. In our experiment, the $conv_{4\_3}$ layer of VGG-16 network [39] is adopted.

At the second stage, we train VCGAN generator and discriminator alternatively and include optical flow for matching spatial location. The overall objective is defined as:

$$L_{2nd} = \lambda_1 L_1 + \lambda_p L_p + \lambda_G L_G + \lambda_{st} L_{st} + \lambda_{dlt} L_{dlt}, \tag{8}$$

where L_G, L_{st} and L_{dlt} indicate GAN loss, short-term loss, and dense long-term loss, respectively.

For the GAN training process, WGAN critic [42] is utilized, which is defined as:

$$L_G = -\mathbb{E}[D(s_t)], \tag{9}$$

$$L_D = \mathbb{E}[D(s_t)] - \mathbb{E}[D(z)], \tag{10}$$

where equation (9) and (10) constitute WGAN loss for generator $G(*)$ and discriminator $D(*)$, respectively. Due to spectral normalization [43] attached to each convolutional layer of discriminator, VCGAN satisfies the 1-Lipschitz continuity.

To enforce temporal consistency, VCGAN should also learn connections for continuously generated frames. Suppose that there are N continuous frames used for training in each iteration, the optical flow-based objectives include short-term loss and dense long-term loss are defined as:

$$L_{st} = \mathbb{E}[\sum_{t=2}^{T} M_{t-1\to t} ||s_t - W(O_{t-1\to t}, s_{t-1})||_1], \tag{11}$$

$$L_{dlt} = \mathbb{E}[\sum_{t=3}^{T} \sum_{m=1}^{t-2} M_{m\to t} ||s_t - W(O_{m\to n}, s_m)||_1], \tag{12}$$

where T is the length of frames, s_m and s_t are the colorized frames at time m and t, respectively. $M_{m\to t}$ and $O_{m\to t}$ represent the non-occlusion mask [28] and real forward flow of colorful images between time m and t, respectively. The operator $W(*)$ warps input frame under the guidance of flow $O_{m\to t}$. By matching the pixel-wise non-occlusion region of the warped frame and current output, it enforces the temporal consistency of correctly warped regions. The short-term loss learns the color similarity for neighbouring frames. The dense long-term loss models each remote connection between two generated frames. Moreover, we follow the protocol in [28] to estimate mask $M_{m\to t} = exp(-\alpha||x_t, W(O_{m\to t}, x_m)||_2^2)$, indicating the clean regions of warped image. The scale factor α enlarges the numerical disparity between occlusion and non-occlusion regions.

IV. Experiment

A. Implementation Details

Dataset. To enhance the performance of VCGAN, we use the entire ImageNet [33] dataset (1281167 images with 1000 categories) at the first training stage. The images are resized to 256×256. The images encoded as grayscale are excluded. At the second training stage, we utilize the DAVIS [34] and Videvo [35] datasets that contain 156 short videos (overall 29620 images). We assume each short video is equally important. All training images are normalized to the range of [-1, 1].

Network. Both the generator and discriminator adopt LeakyReLU [44] activation function. The instance normalization [45] is attached to each convolutional layer of both encoder and discriminator except the first and the last layers. Note that, the pre-trained ResNet-50-IN [31] also adopts LeakyReLU [44] activation function and instance normalization [45]. Specifically, to maintain more information while performing the down-sampling operation, the pooling layers of the original ResNet-50-IN architecture [31], [45] are replaced by convolutional layers with a stride of 2. At the final part of the network, an additional convolutional layer is added to reduce the dimension from 2048 to 512. We train this ResNet-50-IN from scratch following the hyper-parameter settings of [31] until the ImageNet validation accuracy is high enough and stable. Then, the weights are loaded to the two feature extractors of VCGAN, while the weights of other layers of VCGAN are Xavier initialized [46].

Optimization. For the first stage, the generator of VCGAN is trained with equation (5) for 20 epochs while the learning

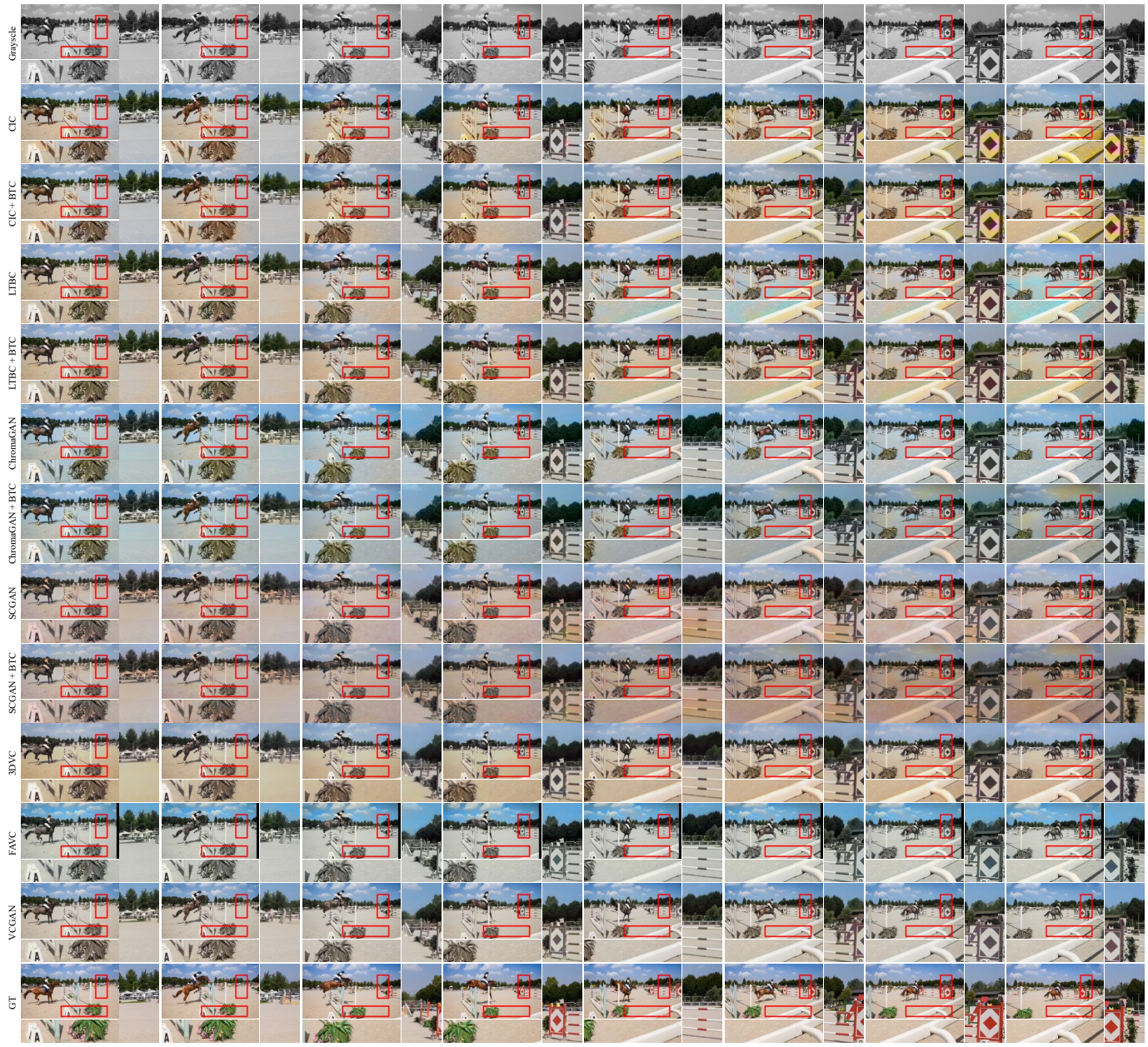


Fig. 4. Colorization comparison on "horsejump-high" from DAVIS [34] dataset. The first and last row include the grayscale and colorful ground truth frames, respectively. The middle rows include colorized results from state-of-the-art methods CIC [4], CIC + BTC [4], [28], LTBC [5], LTBC + BTC [5], [28], FAVC [14] and the proposed VCGAN. The red rectangles highlight the inconsistent regions or strange colors for the baselines. The shown eight frames correspond to the location 0th, 7th, 14th, 21st, 28th, 35th, 42nd, and 49th of original video, respectively. The interval of each two frames is 7. Please refer to supplementary materials for more results (representative video clips).

rate is initialized to 2×10^{-4}, which is halved after 10 epochs. For the second stage, we load the weights from the first stage for VCGAN generator. Then, the whole VCGAN is optimized using equation (8) on 256p resolution and 480p resolution, for 500 epochs and 500 epochs, respectively. The initial learning rates for both generator and discriminator equal to 5×10^{-5}. For 480p resolution, the learning rate is halved every 100 epochs. For a single category, we randomly sample $T = 5$ successive frames at one iteration. The scale factor α of the non-occlusion mask is set to 50. For the optimization, we use Adam optimizer [47] with $\beta_1 = 0.5$, $\beta_2 = 0.999$. The batch size equals to 16 and 4 for the two stages, respectively.

The coefficients λ_1, λ_p, λ_G, λ_{st}, λ_{dlt} are empirically set to 10, 5, 1, 3, 5, respectively. At the first stage, the VCGAN is trained on 4 NVIDIA Titan Xp GPUs. At the second stage, the training processes on 256p resolution and 480p resolution are performed on 4 NVIDIA Titan Xp GPUs (12 Gb) and 4 NVIDIA Tesla V100 GPUs (32 Gb), respectively. We implement the VCGAN using the PyTorch 1.0.0 framework with Cuda 8.0, Python 3.6. The whole training of VCGAN takes approximately 14 days (including 10, 1, and 3 days for the first stage, second stage on 256p and 480p, respectively).

B. Experiment Settings

Dataset. Following the settings of [14], [28], we perform the evaluations on DAVIS [34] and Videvo [35] testing set.

Fig. 5. Colorization comparison on "SkateboarderTableJump" from Videvo [35] dataset. The first and last row include the grayscale and colorful ground truth frames, respectively. The middle rows include colorized results from state-of-the-art methods CIC [4], CIC + BTC [4], [28], LTBC [5], LTBC + BTC [5], [28], FAVC [14] and the proposed VCGAN. The red rectangles highlight the inconsistent regions or strange colors for the baselines. The shown eight frames correspond to the location 0th, 40th, 80th, 90th, 100th, 105th, 110th, and 119th of original video, respectively. The interval of first two frames is 40.

The DAVIS dataset includes 30 short videos, each of which contains approximately 100 frames. The Videvo dataset consists of 20 videos and there are about 300 frames in each clip. Although different approaches may produce images of diverse resolutions, all the result images are generated and resized to match the image resolution of ground truth for fair. Moreover, since the proposed VCGAN can generate a single colorful image using weights of the first stage, we assess its colorization quality by colorizing single images. We select the 10000 ImageNet validation images [33] based on the settings in [2], [4], [11], [12].

PSNR and SSIM [48]. To represent the fidelity of generated image, we apply PSNR to calculate the pixel-level error. Since PSNR is not highly relevant to the human visual system, we also adopt SSIM [48] to estimate the structural similarity (especially luminance, contrast, structure).

Top-5 Accuracy. To estimate the semantic interpretability, we adopt the Top-5 Accuracy. It is only for evaluating image colorization quality based on a pre-trained VGG-16 network.

Warp Error. For video colorization, the temporal continuity of generated frames is equally significant with colorization quality. We measure the spatiotemporal consistency by computing the disparity between every warped previous frame and current frame. The warp error of one video is defined as:

$$WE = \sum_{t=2}^{T} \frac{hw}{hw - sum(M_t)} M_t ||v_t - W(O_{t-1\to t}, v_{t-1})||_2^2, \tag{13}$$

where v_t is generated frame at time t and $W(O_{t-1\to t}, v_{t-1})$ is the warped frame from previous frame. It is weighted by the

TABLE I
COMPARISON OF VIDEO COLORIZATION METHODS [4], [5], [11], [12], [14], [15], [28] AND THE PROPOSED VCGAN ON DAVIS AND VIDEVO DATASETS. THE **RED**, **BLUE**, AND GREEN COLORS REPRESENT THE BEST, THE SECOND-BEST, AND THE THIRD-BEST PERFORMANCES, RESPECTIVELY.

Method	DAVIS			Videvo			Network Architecture		
	PSNR	SSIM	Warp Error	PSNR	SSIM	Warp Error	Semantic Model	Flow Estimator	Hybrid Model
Grayscale	23.77	0.9484	/	25.31	0.9570	/	/	/	/
CIC [4]	22.44	0.9003	0.06055	21.79	0.8989	0.03317	✓	/	/
LTBC [5]	23.89	0.9130	0.05901	24.64	0.9237	0.03285	✓	/	/
ChromaGAN [11]	23.70	0.9377	0.06023	23.88	0.9354	0.03319	✓	/	/
SCGAN [12]	23.19	0.8959	0.05918	23.29	0.8549	0.03301	✓	/	/
CIC + BTC [28]	21.48	0.8898	0.05170	21.02	0.8800	0.02891	✓	FlowNet2	/
LTBC + BTC [28]	22.45	0.9006	0.05144	22.81	0.9072	0.02995	✓	FlowNet2	/
ChromaGAN + BTC [28]	19.88	0.8896	**0.04955**	16.63	0.8289	0.02753	✓	FlowNet2	/
SCGAN + BTC [28]	19.35	0.8716	**0.04902**	16.28	0.7455	**0.02715**	✓	FlowNet2	/
3DVC [15]	**23.43**	**0.9115**	0.05125	**24.28**	**0.9200**	**0.02659**	/	3D Conv	/
FAVC [14]	**22.98**	**0.9055**	0.06002	**23.47**	**0.9183**	0.03236	✓	PWC-Net	/
VCGAN	**23.77**	**0.9196**	**0.04871**	**25.11**	**0.9264**	**0.02502**	✓	PWC-Net	✓

number of occlusion pixels. Note that, M_t is a binary mask that considers both occlusion regions and motion boundaries. The hw is the overall number of pixels in a frame. For calculation details, we follow the protocol in [49].

C. Video Colorization Comparisons

We compare VCGAN with state-of-the-art video colorization algorithms FAVC [14], 3DVC [15] and 4 representative image colorization methods CIC [4], LTBC [5], ChromaGAN [11], and SCGAN [12]. In addition, we also compare with the task-independent approach BTC [28], which refines the single image colorization results. Thus, there are 6 video colorization results (i.e., FAVC, 3DVC, CIC + BTC, LTBC + BTC, ChromaGAN + BTC, and SCGAN + BTC) in the experiment. The training sets of the baselines are the same to VCGAN (i.e., ImageNet [33], DAVIS [34], and Videvo [35]).

The quantitative comparison on 480p validation sets is concluded in Table I. The results of grayscale frames serve as baseline. Also, we do not include single image colorization methods [4], [5], [11], [12] in comparisons since they do not count the temporal continuity. Since the BTC strikes a balance between colorization quality and temporal coherence, the disparity between neighbouring frames is much smaller (e.g., the Warp Error of CIC + BTC is much smaller than CIC itself). However, the results from CIC + BTC may suffer from a decrease of PSNR, since the frame-wise characteristic may be weaken. The FAVC predicts four perceptually satisfactory colorizations, it may sacrifice good performance on metrics. The 3DVC adopts 3D Conv to learn inter-frame relations and colorization jointly. Therefore, it achieves better results than FAVC. The proposed VCGAN framework achieves the best pixel fidelity (PSNR, SSIM) and spatiotemporal consistency (Warp Error) among all the video colorization methods. It demonstrates that the proposed two feature extractors and dense long-term loss L_{dlt} are obviously beneficial to video colorization. The proposed VCGAN uses semantic model (i.e., two feature extractors) like the baselines except 3DVC, which promotes fast convergence and high colorization quality. In addition, **the VCGAN is the only hybrid model that unifies both image and video colorization in same architecture.**

The qualitative samples are shown in Figure 4 and 5. As shown in the highlighted patches (by red rectangles), the image colorization methods CIC, LTBC, ChromaGAN, and SCGAN are not temporally consistent enough. It makes their temporally smoothed results by BTC also not very continuous since their training of image colorization and temporal smoothing are separated. For instance, in Figure 4, the ground is colorized in yellow by CIC and blue by LTBC. Also, in Figure 5, the arms are colorized in red by CIC and blue by LTBC. Compared with them, the results from 3DVC, FAVC and the proposed VCGAN are much more continuous. However, the color tones of 3DVC and FAVC results may be not very realistic. For instance, in both Figure 4 and 5, the objects (e.g., the sky, horse, and man) colorized by FAVC are dusky. The colors of 3DVC results are not as natural as VCGAN. Compared with aforementioned methods, the proposed VCGAN produces plausible and temporally coherent frames. We will include more comparison samples in supplementary materials.

D. Ablation Study

To discover the effectiveness of different loss terms, feature extractors, and the proposed training scheme used in VCGAN, We conduct several experiments as an ablation study. The ablation studies are performed on same datasets, i.e., DAVIS [34] and Videvo [35] 480p validation data. There are 20 settings with abbreviations as concluded in Table II.

Loss terms. The settings l(1) and l(2) (i.e., VCGAN only trained with L1 loss or joint L1 loss and short-term loss) serve as baselines. The settings l(3.1)-l(3.7) and l(4.1)-l(4.5) are designed to evaluate "colorization reality" (e.g., without L_1, L_p and L_G) and "smoothing ability" (e.g., without L_{st} and L_{dlt}), respectively. However, different loss terms have internal relations since the VCGAN is trained by the combinations of the losses with individual coefficients. For instance, if we drop the perceptual loss L_p of VCGAN, the output frames may be smoother than trained with full losses (i.e., Warp Error is smaller). It is because the terms L_{st} and L_{dlt} account relatively more coefficients in this setting than trained with full losses. Thus, we suggest readers **compare the PSNR and SSIM for "colorization quality"-related settings (e.g., without L_p)** since they may care more about pixel-level accuracy. Similarly, please **focus on the Warp Error for "smoothing ability"-related settings (e.g., without L_{st})**. The evaluation results are concluded in Table III.

TABLE II
THE CONCLUSION OF ALL ABLATION STUDY SETTINGS, WHERE "/" DENOTES "NO CHANGE".

Setting	Loss terms	Feature extractors	Training scheme	Target
l(1)	L_1	/	/	only adopting L1 loss as a baseline
l(2)	L_1, L_{st}	/	/	using L1 and short-term loss to meet Markov Chain as a baseline
l(3.1)	L_p, L_G, L_{st}, L_{dlt}	/	/	w/o L_1 to evaluate its effect on colorization quality
l(3.2)	L_1, L_G, L_{st}, L_{dlt}	/	/	w/o L_p to evaluate its effect on colorization quality
l(3.3)	L_1, L_p, L_{st}, L_{dlt}	/	/	w/o L_G to evaluate its effect on colorization quality
l(3.4)	L_1, L_{st}, L_{dlt}	/	/	w/o L_p, L_G to evaluate their effects on colorization quality
l(3.5)	L_p, L_{st}, L_{dlt}	/	/	w/o L_1, L_G to evaluate their effects on colorization quality
l(3.6)	L_G, L_{st}, L_{dlt}	/	/	w/o L_1, L_p to evaluate their effects on colorization quality
l(3.7)	L_{st}, L_{dlt}	/	/	w/o L_1, L_p, L_G to evaluate their effects on colorization quality
l(4.1)	L_1, L_p, L_G, L_{dlt}	/	/	w/o L_{st} to evaluate its effect on smoothing ability
l(4.2)	L_1, L_p, L_G, L_{st}	/	/	w/o L_{dlt} to evaluate its effect on smoothing ability
l(4.3)	L_1, L_p, L_G	/	/	w/o L_{st}, L_{dlt} to evaluate their effects on smoothing ability
l(4.4)	L_1, L_p, L_G, L_{lt}	/	/	evaluating L_{dlt} by only adopting normal long-term loss L_{lt}
l(4.5)	L_1, L_p, L_G, L_{st}, L_{lt}	/	/	evaluating L_{dlt} by replacing it with normal long-term loss L_{lt}
f(1)	/	w/o GFE	/	w/o global feature extractor
f(2)	/	w/o PFE	/	w/o placeholder feature extractor and recurrent connection from s_{t-1}
f(3)	/	w/o GFE, PFE	/	w/o both feature extractors and recurrent connection from s_{t-1}
t(1)	/	/	1st stage	adopting VCGAN first stage model, where only L_1 and L_p used
t(2)	/	/	2nd stage, 256p	adopting VCGAN second stage model, but trained on 256p data
VCGAN	L_1, L_p, L_G, L_{st}, L_{dlt}	/	**2nd stage, 480p**	**VCGAN model trained with full loss terms and training stages**

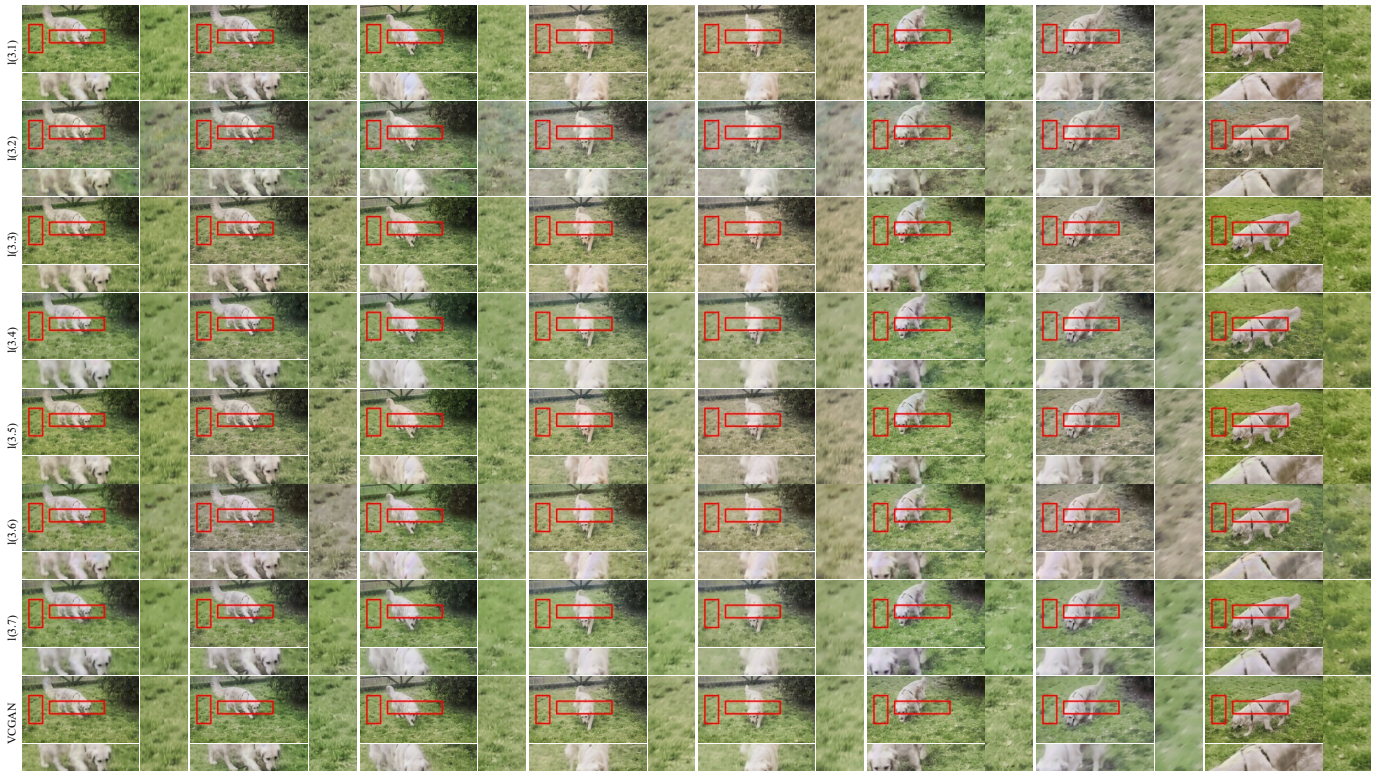


Fig. 6. The comparison of colorization quality on "dog" from DAVIS [34] dataset. The shown eight frames correspond to the location 0th, 1st, 9th, 18th, 20th, 34th, 39th, and 59th of original video, respectively. The average interval of two frames is larger than 5.

Loss terms related to colorization quality. As shown in Table III, the baseline setting l(1) obtains the worst result since it only uses L_1 for training. By adding the short-term loss L_{st} (setting l(2)), the VCGAN has much better results since it meets the Markov Chain assumption. For the settings l(3.1)-l(3.3), if VCGAN trained without L_1, L_p, or L_G, there is an obvious drop in terms of PSNR and SSIM metrics, which demonstrate that all of them are beneficial for colorization quality. As shown in Figure 6, L_p or L_G promotes VCGAN to generate more realistic and vivid colorizations. Similarly, for the settings l(3.4)-l(3.7), these two metrics are worse than full loss terms, since more than one "colorization quality"-related losses are removed. Furthermore, the results (e.g., the grass and dog) of l(3.1)-l(3.7) are also of poor color contrast, as shown in Figure 6, which demonstrates that L_1, L_p, or L_G are vital for VCGAN to produce high-quality colorizations.

Loss terms related to smoothing ability. The most significant presumption for video generation is that the produced frames satisfy Markov Chain. Since the setting l(4.1) does not adopt L_{st}, it obtains higher Warp Error (e.g., 0.00433

TABLE III
VCGAN ABLATION STUDY ON DAVIS AND VIDEVO DATASETS. SINCE DIFFERENT WEIGHTS OF LOSS TERMS AFFECT EACH OTHER, PLEASE COMPARE THE PSNR AND SSIM FOR **COLORIZATION QUALITY** SETTINGS AND THE WARP ERROR FOR **SMOOTHING ABILITY** SETTINGS BASED ON **FULL VCGAN**. PLEASE COMPARE THE ALL METRICS FOR **BASELINE**, **FEATURE EXTRACTOR**, AND **TRAINING SCHEME** SETTINGS WITH **FULL VCGAN**. PLEASE COMPARE THE **HIGHLIGHTED ITEMS** IN EACH GROUP, WHERE **RED** COLOR DENOTES THE BEST PERFORMANCE.

Setting	Method	Target	DAVIS			Videvo		
			PSNR	SSIM	Warp Error	PSNR	SSIM	Warp Error
l(1)	L_1	**baseline**	23.64	0.9138	0.05495	24.84	0.9250	0.02872
l(2)	L_1, L_{st}	**baseline**	23.57	0.9017	0.05330	24.96	0.9159	0.02773
l(3.1)	L_p, L_G, L_{st}, L_{dlt}	**colorization quality**	**23.71**	**0.9139**	0.05218	**25.16**	**0.9255**	0.02727
l(3.2)	L_1, L_G, L_{st}, L_{dlt}	**colorization quality**	**23.02**	**0.8489**	0.04688	**24.12**	**0.8679**	0.02417
l(3.3)	L_1, L_p, L_{st}, L_{dlt}	**colorization quality**	**23.77**	**0.9141**	0.05221	**25.18**	**0.9256**	0.02735
l(3.4)	L_1, L_{st}, L_{dlt}	**colorization quality**	**22.77**	**0.8264**	0.04611	**24.03**	**0.8509**	0.02332
l(3.5)	L_p, L_{st}, L_{dlt}	**colorization quality**	**23.74**	**0.9135**	0.05193	**25.09**	**0.9253**	0.02718
l(3.6)	L_G, L_{st}, L_{dlt}	**colorization quality**	**23.21**	**0.8519**	0.04783	**24.11**	**0.8719**	0.02441
l(3.7)	L_{st}, L_{dlt}	**colorization quality**	**22.81**	**0.8257**	0.04649	**23.95**	**0.8492**	0.02329
VCGAN	**full losses and training stages**	**full VCGAN**	**23.77**	**0.9196**	0.04871	**25.11**	**0.9264**	0.02502
l(4.1)	L_1, L_p, L_G, L_{dlt}	**smoothing ability**	23.85	0.9147	**0.05304**	24.96	0.9246	**0.02748**
l(4.2)	L_1, L_p, L_G, L_{st}	**smoothing ability**	23.81	0.9147	**0.05270**	25.15	0.9263	**0.02777**
l(4.3)	L_1, L_p, L_G	**smoothing ability**	23.61	0.9139	**0.05373**	24.92	0.9252	**0.02818**
l(4.4)	L_1, L_p, L_G, L_{lt}	**smoothing ability**	23.66	0.9129	**0.05203**	24.79	0.9238	**0.02740**
l(4.5)	L_1, L_p, L_G, L_{st}, L_{lt}	**smoothing ability**	23.65	0.9147	**0.05371**	25.10	0.9254	**0.02840**
VCGAN	**full losses and training stages**	**full VCGAN**	23.77	0.9196	**0.04871**	25.11	0.9264	**0.02502**
f(1)	w/o GFE	**feature extractors**	**23.47**	**0.9135**	**0.05203**	**25.00**	**0.9251**	**0.02778**
f(2)	w/o PFE	**feature extractors**	**23.38**	**0.9122**	**0.05170**	**25.02**	**0.9249**	**0.02761**
f(3)	w/o GFE, PFE	**feature extractors**	**23.21**	**0.9130**	**0.05232**	**24.91**	**0.9241**	**0.02826**
t(1)	1st training stage, ImageNet	**training scheme**	**23.53**	**0.9137**	**0.05559**	**24.59**	**0.9228**	**0.02918**
t(2)	2nd training stage, 256p videos	**training scheme**	**23.65**	**0.9095**	**0.05128**	**24.27**	**0.9179**	**0.02664**
VCGAN	**full losses and training stages**	**full VCGAN**	**23.77**	**0.9196**	**0.04871**	**25.11**	**0.9264**	**0.02502**

Fig. 7. The comparison of smoothing ability on "cows" from DAVIS [34] dataset. The shown eight frames correspond to the location 50th, 51st, 52nd, 53rd, 54th, 55th, 56th, and 57th of original video, respectively. The interval of each two frames is 1.

and 0.00246 increases on DAVIS and Videvo, respectively). Similarly, the Warp Error increases if removing L_{dlt} (i.e., l(4.2)) or both L_{st} and L_{dlt} (i.e., l(4.3)). As shown in Figure 7, the colorized frames from l(4.1)-l(4.3) are not continuous enough, i.e., the colors of continuous frames are not consistent. For the settings l(4.4) and l(4.5), we replace the proposed dense long-term loss L_{dlt} with normal long-term loss L_{lt} [28], which only panels the differences between current frame and the first frame. The Warp Errors of these settings are still inferior to full VCGAN.

Dense long-term loss L_{dlt}. Some previous methods set the time range equals to 2 (previous and current frame) [14], [15], [18], [25], [50] or 3 (previous, current, and leading frames) [49], [51]. They did not consider the long-term or remote relations. Lai *et al.* [28] incorporated a long-term loss L_{lt} modeling the connection of current frame and the first frame. However, the proposed dense long-term loss L_{dlt} includes each remote correlation between current frame and all previous generated frames. To demonstrate its effectiveness, we fix the "colorization quality"-related losses L_1, L_p, and L_G and use one additional loss from L_{st}, L_{lt}, and L_{dlt}, i.e., l(4.2), l(4.4), and l(4.1). In terms of Warp Error, L_{st} (l(4.1)) is the

Fig. 8. The illustration of utilization of the proposed dense long-term loss L_{dlt} on "Ducks" from Videvo [35] dataset. The shown eight frames correspond to the location 11st, 12nd, 22nd, 23rd, 115th, 116th, 251st, and 252nd of original video, respectively. The last two frames are very remote in terms of the first frame.

Fig. 9. The comparison of feature extractors and training scheme on "YogaHut2" from Videvo [35] dataset. The shown eight frames correspond to the location 0th, 10th, 20th, 30th, 40th, 50th, 60th, and 70th of original video, respectively.

most significant factor to smooth videos since it panels the neighbouring frames. Though L_{dlt} (l(4.2)) does not contain the consistency of neighbouring frames, it panels each remote frames to minimize the color differences. Compared with L_{lt} (l(4.4)), the L_{dlt} (l(4.2)) achieves lower Warp Error, which demonstrates that modeling all remote relations are beneficial to enhance temporal consistency.

In addition, we add a setting l(4.5) that replaces L_{dlt} with L_{lt}. As shown in Table III full VCGAN setting, L_{dlt} explicitly reduces Warp Errors by approximately 0.00500 and 0.00338 on DAVIS and Videvo datasets, respectively, than L_{lt}. In addition, since the continuous frames may not represent long-term consistency, we illustrate remote frames in Figure 8 to show the effect of the proposed dense long-term loss L_{dlt}. Only the VCGAN trained with L_{dlt} produces consistent background color (i.e., blue sky); whereas the normal long-term loss L_{lt} fails to maintain the consistency for remote frames. In all settings, VCGAN with full losses better balances colorization fidelity and spatiotemporal constancy. However, other settings will one-sidedly emphasize PSNR or warp error, which demonstrates each loss term is significant for VCGAN.

Feature extractors. To demonstrate the advance of the proposed two feature extractors, we remove the global feature extractor (GFE) or placeholder feature extractor (PFE) or both for comparisons (i.e., f(1), f(2), and f(3)). The GFE is a pre-trained ResNet-50-IN, which provides semantics for the VCGAN to identify colors for objects with similar edges [12]. Therefore, f(1) obtains worse PSNR and SSIM values. Also, we found the Warp Errors of f(1) are higher than full VCGAN, which proves that the semantics provided by the pre-trained GFE are also beneficial to minimize inter-frame disparity. For f(2), it proves that the PFE can provide the information from last colorized frame. Otherwise, the Warp Error increases due to no use of the PFE with recurrent connection. For f(3), it obtains worse results since only the mainstream of VCGAN is used. As shown in Figure 9, the patches are less colorful

Fig. 10. The comparison of VCGAN trained with different objective coefficients on "Surfing" from Videvo [35] dataset. The shown eight frames correspond to the location 0th, 10th, 50th, 70th, 80th, 110th, 130th, and 131st of original video, respectively.

TABLE IV
THE EXPERIMENT CONCLUSION OF THE SENSITIVENESS OF LOSS COEFFICIENTS. THE **RED**, **BLUE**, AND GREEN COLORS REPRESENT THE BEST, THE SECOND-BEST, AND THE THIRD-BEST PERFORMANCES, RESPECTIVELY.

Setting	λ_1	λ_p	λ_G	λ_{st}	λ_{dlt}	Target	DAVIS			Videvo		
							PSNR	SSIM	Warp Error	PSNR	SSIM	Warp Error
s(1)	1	1	1	1	1	all "1" coefficients	**23.83**	**0.9193**	0.05101	24.68	0.9224	0.02659
s(2)	20	5	1	3	5	double λ_1	**23.90**	0.9192	0.05042	**25.11**	**0.9244**	0.02746
s(3)	10	10	1	3	5	double λ_p	**23.85**	**0.9202**	0.04971	**25.20**	0.9232	0.02602
s(4)	10	5	2	3	5	double λ_G	23.32	0.9113	0.04957	24.66	0.9211	0.02644
s(5)	10	5	1	6	5	double λ_{st}	23.75	0.9133	**0.04915**	24.55	0.9197	0.02565
s(6)	10	5	1	3	10	double λ_{dlt}	23.37	0.9096	**0.04909**	24.67	0.9194	**0.02536**
s(7)	20	10	2	3	5	double λ_1, λ_p, and λ_G	23.63	0.9118	0.04933	**25.07**	**0.9245**	0.02650
s(8)	10	5	1	6	10	double λ_{st} and λ_{dlt}	23.76	0.9127	**0.04871**	24.56	0.9199	**0.02501**
VCGAN	**10**	**5**	**1**	**3**	**5**	**full VCGAN**	23.77	**0.9196**	**0.04871**	**25.11**	**0.9264**	**0.02502**

than full VCGAN.

Training scheme. For the proposed training scheme, we include the VCGAN first and second training stage (on 256p resolution) models for comparisons (i.e., t(1) and t(2)). Since the image resolution and loss terms (e.g., temporal losses) are both different from the full VCGAN, directly applying first stage model leads to extremely inconsistent videos. Similarly, if the training resolution and testing resolution are unequal, the result is not plausible, e.g., the VCGAN trained on 256p data. Some results are shown in Figure 9, where the colors are not as vivid as full VCGAN and the frames are not continuous enough compared with full VCGAN.

In conclusion, each component is vital for the proposed VCGAN to obtain high-quality and temporally smooth video colorizations. Also, the proposed dense long-term loss further ensures the consistency of far frames.

E. Investigation of the Sensitiveness of Loss Coefficients

The coefficients of the objectives used for VCGAN optimization are empirically selected. To demonstrate that the proposed values are relatively better than other combinations, we conduct several experiments by adjusting some of the coefficients. The results on DAVIS and Videvo datasets are in Table IV. The proposed coefficients achieve relatively better values in terms of PSNR, SSIM, and Warp Error metrics. Also as shown in Figure 10, if VCGAN trained with all "1" coefficients, the colors are very consistent for far frames, since it may not balance high-quality colorization and temporal consistency well. If doubling λ_G (i.e., s(4)), the results are almost monochrome.

F. Image Colorization Results

If the input for the placeholder feature extractor is replaced with a grayscale image, the proposed VCGAN turns into an image colorization model (i.e., a video only contains one frame). To demonstrate the image colorization ability of VCGAN, we compare the VCGAN first training stage model with 7 state-of-the-art image colorization algorithms [3]–[5], [11], [14], [52], where the colorization part of [14] is adopted. Note that, the training sets of the methods are the same (i.e., ImageNet [33]). Following the settings in [4], we choose the 10000 images from the ImageNet validation set for evaluation.

We illustrate some colorized results in Figure 11. There are obvious visual artifacts in some of the methods, as shown in Figure 11. For instance, there is a color bleeding artifact (i.e., the color of one object permeates to other objects) in row 1-4 of CIC [4] and row 3, 4 of DeOldify [52]. Even though there is no color bleeding of FAVC [14], FAVC results are not colorful enough compared with other methods. However, the results generated by VCGAN are more colorful and reasonable than other methods. Also, there are almost no artifacts in the results. In conclusion, the hybrid VCGAN architecture is appropriate for both image and video colorization tasks.

The quantitative analysis is summarized in Table V. The proposed VCGAN achieves the best PSNR. It demonstrates that the VCGAN architecture produces the colorizations with the highest pixel fidelity. Also, it obtains the second-best SSIM and Top-5 Accuracy metrics, which evaluates the semantic representation ability of colorization systems. It demonstrates that the VCGAN architecture can generate relatively more plausible colorizations than other methods. The GAN-based methods (Pix2Pix [3], DeOldify [52], ChromaGAN [11] and the proposed VCGAN) obtain better performance, since the

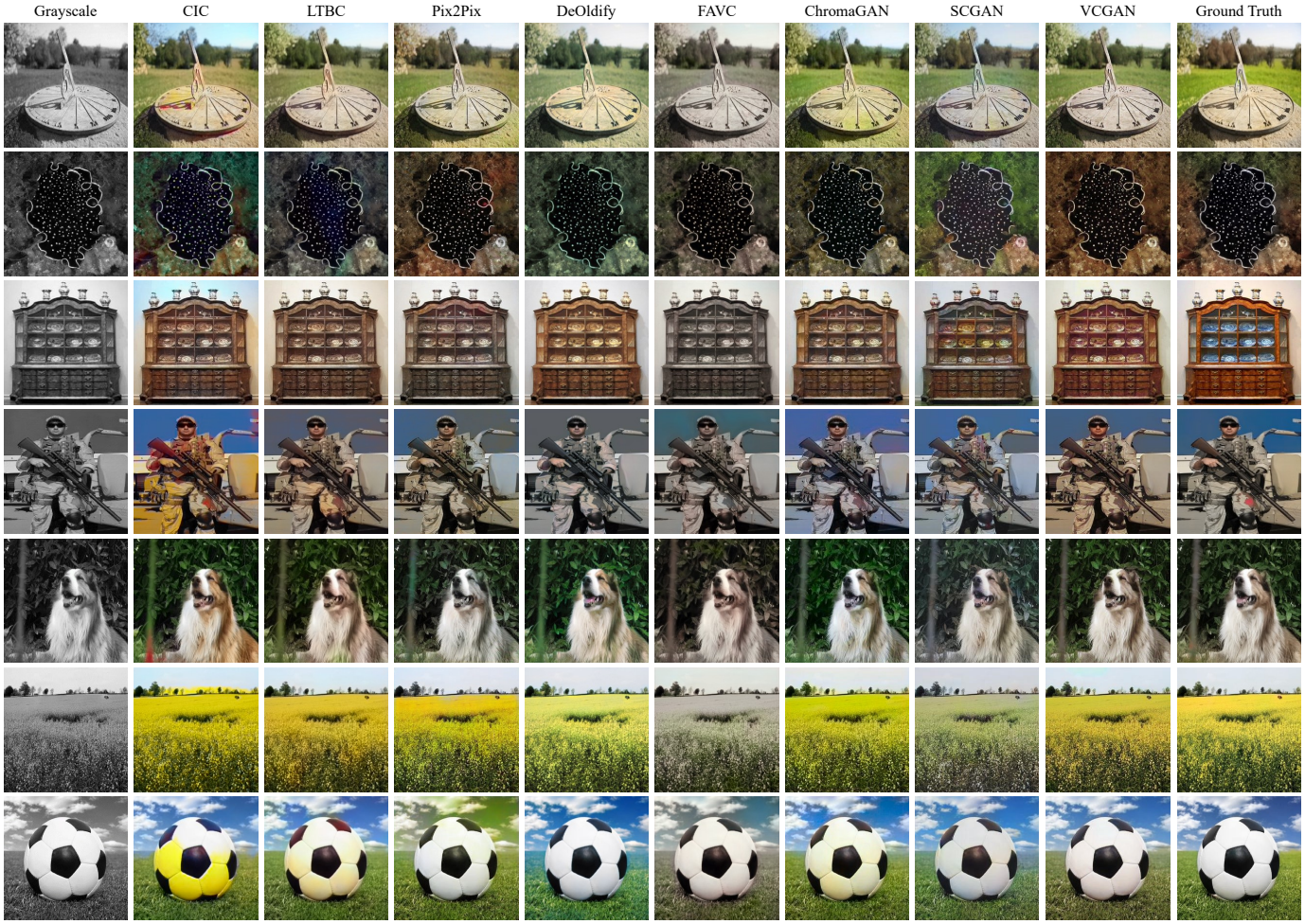


Fig. 11. Illustration of image colorization results of VCGAN (first stage) and state-of-the-art methods [3]–[5], [11], [12], [14], [52] on ImageNet validation set. The first column and last column denote the grayscale and colorful ground truth. The other columns include the colorizations of the methods in the experiment. The red rectangles in the figures represent inconsistent regions or strange colors.

TABLE V
COMPARISON OF STATE-OF-THE-ART IMAGE COLORIZATION METHODS [3]–[5], [11], [12], [14], [52] AND PROPOSED VCGAN (FIRST STAGE). THE **RED**, **BLUE**, AND GREEN COLORS REPRESENT THE BEST, THE SECOND-BEST, AND THE THIRD-BEST PERFORMANCES, RESPECTIVELY.

Method	PSNR	SSIM	Top-5 Acc	GAN Training
Ground Truth	/	1	84.91%	/
Grayscale	23.23	0.9394	73.81%	/
CIC [4]	22.49	0.9153	**78.11%**	/
LTBC [5]	**24.32**	**0.9464**	77.13%	/
Pix2Pix [3]	23.25	0.9386	76.57%	✓
DeOldify [52]	23.14	0.9194	78.01%	✓
FAVC [14]	22.96	0.9146	76.76%	/
ChromaGAN [11]	**24.32**	0.9273	**78.51%**	✓
SCGAN [12]	**23.80**	**0.9470**	76.70%	✓
VCGAN	**24.48**	**0.9427**	**78.19%**	✓

GAN facilitates sharper results, which are difficult to accomplish by only adopting L1 loss.

G. Failure Cases and Discussion

The proposed VCGAN produces relatively plausible colorful videos in many cases. However, there still exists a common issue when there are a lot of details in each frame (left part of Figure 12). Also, since the video colorization is an ill-posed problem, the produced frames maybe not colorful enough (right part of Figure 12). The more complicated video training datasets may enhance the performance of VCGAN. In the future, we will further improve VCGAN architecture to make it faster and produce more plausible and colorful results.

V. CONCLUSION

In this paper, we presented a recurrent VCGAN framework for automatically generating photorealistic and temporally coherent video colorization. Utilizing two pre-trained ResNet-50-IN networks as the global feature and placeholder feature extractors along with the U-Net-based mainstream, it extracts semantics efficiently while maintains the spatiotemporal consistency among consecutive frames recurrently. By changing the input for placeholder feature extractor, VCGAN architecture unifies both image and video colorization applications. Furthermore, the proposed dense long-term loss models each remote relations of continuous frames. It enhances the smoothness of generated videos while requires ignorable additional memory. Finally, we validated VCGAN with several state-of-the-art image and video colorization methods. The experiment

Fig. 12. Failure cases of VCGAN. The rows from top to bottom denote the grayscale input, colorized frames by VCGAN and ground truth, respectively.

results demonstrate that the proposed VCGAN obtains better performances in both tasks than the well-known methods.

ACKNOWLEDGMENT

The authors would like to thank Bei Li, Pengfei Xian, Xuihui Wang and Wei Liu for many helpful comments. The authors would also like to thank the anonymous reviewers and the editors for their kind suggestions.

REFERENCES

[1] Z. Cheng, Q. Yang, and B. Sheng, "Deep colorization," in *Proc. ICCV*, 2015, pp. 415–423.
[2] G. Larsson, M. Maire, and G. Shakhnarovich, "Learning representations for automatic colorization," in *Proc. ECCV*, 2016, pp. 577–593.
[3] 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.
[4] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," in *Proc. ECCV*, 2016, pp. 649–666.
[5] 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.
[6] A. Deshpande, J. Lu, M.-C. Yeh, M. Jin Chong, and D. Forsyth, "Learning diverse image colorization," in *Proc. CVPR*, 2017, pp. 6837–6845.
[7] A. Royer, A. Kolesnikov, and C. H. Lampert, "Probabilistic image colorization," in *Proc. BMVC*, 2017, pp. 85.1–85.12.
[8] Y. Cao, Z. Zhou, W. Zhang, and Y. Yu, "Unsupervised diverse colorization via generative adversarial networks," in *Proc. ECMLPKDD*, 2017, pp. 151–166.
[9] 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.
[10] J. Zhao, J. Han, L. Shao, and C. G. Snoek, "Pixelated semantic colorization," *Int. J. Comput. Vis.*, pp. 1–17, 2019.
[11] P. Vitoria, L. Raad, and C. Ballester, "Chromagan: Adversarial picture colorization with semantic class distribution," in *Proc. WACV*, 2020, pp. 2445–2454.
[12] 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.
[13] J.-W. Su, H.-K. Chu, and J.-B. Huang, "Instance-aware image colorization," in *Proc. CVPR*, 2020, pp. 7968–7977.
[14] C. Lei and Q. Chen, "Fully automatic video colorization with self-regularization and diversity," in *Proc. CVPR*, 2019, pp. 3753–3761.
[15] P. Kouzouglidis, G. Sfikas, and C. Nikou, "Automatic video colorization using 3d conditional generative adversarial networks," in *Proc. ISVC*, 2019, pp. 209–218.
[16] H. Thasarathan, K. Nazeri, and M. Ebrahimi, "Automatic temporally coherent video colorization," in *Proc. CRV*, 2019, pp. 189–194.
[17] V. Jampani, R. Gadde, and P. V. Gehler, "Video propagation networks," in *Proc. CVPR*, 2017, pp. 451–461.
[18] 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.
[19] S. Wan, Y. Xia, L. Qi, Y.-H. Yang, and M. Atiquzzaman, "Automated colorization of a grayscale image with seed points propagation," *IEEE Trans. Multimedia*, vol. 22, no. 7, pp. 1756–1768, 2020.
[20] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Proc. ECCV*, 2016, pp. 694–711.
[21] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. NeurIPS*, 2014, pp. 2672–2680.
[22] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional gans," in *Proc. CVPR*, 2018, pp. 8798–8807.
[23] A. Levin, D. Lischinski, and Y. Weiss, "Colorization using optimization," *ACM Trans. on Graphics*, vol. 23, no. 3, pp. 689–694, 2004.
[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] 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.
[26] 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.
[27] 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, 2015.
[28] 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.
[29] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Van Der Smagt, D. Cremers, and T. Brox, "Flownet: Learning optical flow with convolutional networks," in *Proc. ICCV*, 2015, pp. 2758–2766.
[30] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. MICCAI*, 2015, pp. 234–241.
[31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, 2016, pp. 770–778.
[32] 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.
[33] 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.
[34] 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.
[35] "videvo," https://www.videvo.net.
[36] T. Welsh, M. Ashikhmin, and K. Mueller, "Transferring color to greyscale images," in *Proc. SIGGRAPH*, 2002, pp. 277–280.
[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] X. Liu, L. Wan, Y. Qu, T.-T. Wong, S. Lin, C.-S. Leung, and P.-A. Heng, "Intrinsic colorization," *ACM Trans. on Graphics*, vol. 27, no. 5, pp. 1–9, 2008.
[39] 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
[40] J.-Y. Zhu, R. Zhang, D. Pathak, T. Darrell, A. A. Efros, O. Wang, and E. Shechtman, "Toward multimodal image-to-image translation," in *Proc. NeurIPS*, 2017, pp. 465–476.

[41] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," *arXiv preprint arXiv:1805.08318*, 2018.
[42] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proc. ICML*, 2017, pp. 214–223.
[43] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," in *Proc. ICLR*, 2018. [Online]. Available: https://openreview.net/forum?id=B1QRgziT-
[44] 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.
[45] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," *arXiv preprint arXiv:1607.08022*, 2016.
[46] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. AISTATS*, 2010, pp. 249–256.
[47] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2015. [Online]. Available: https://openreview.net/forum?id=8gmWwjFyLj
[48] 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.
[49] M. Ruder, A. Dosovitskiy, and T. Brox, "Artistic style transfer for videos," in *Proc. GCPR*, 2016, pp. 26–36.
[50] Y. Xie, E. Franz, M. Chu, and N. Thuerey, "tempogan: A temporally coherent, volumetric gan for super-resolution fluid flow," *ACM Trans. on Graphics*, vol. 37, no. 4, p. 95, 2018.
[51] D. Kim, S. Woo, J.-Y. Lee, and I. S. Kweon, "Deep video inpainting," in *Proc. CVPR*, 2019, pp. 5792–5801.
[52] "Deoldify," https://github.com/jantic/DeOldify.

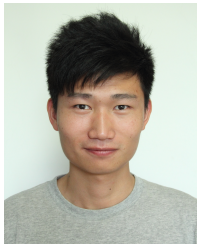
Yuzhi Zhao (S'19) received the B.Eng. Degree in electronic information from Huazhong University of Science and Technology, Wuhan, China, in 2018. He is currently pursuing the Ph.D. degree with the Department of Electronic Engineering, City University of Hong Kong. His research interests include image processing, deep learning and machine learning.

Lai-Man Po (M'92–SM'09) received the B.S. and Ph.D. degrees in electronic engineering from the City University of Hong Kong, Hong Kong, in 1988 and 1991, respectively. He has been with the Department of Electronic Engineering, City University of Hong Kong, since 1991, where he is currently an Associate Professor of Department of Electronic Engineering. He has authored over 150 technical journal and conference papers. His research interests include image and video coding with an emphasis deep learning based computer vision algorithms.

Dr. Po is a member of the Technical Committee on Multimedia Systems and Applications and the IEEE Circuits and Systems Society. He was the Chairman of the IEEE Signal Processing Hong Kong Chapter in 2012 and 2013. He was an Associate Editor of HKIE Transactions in 2011 to 2013. He also served on the Organizing Committee, of the IEEE International Conference on Acoustics, Speech and Signal Processing in 2003, and the IEEE International Conference on Image Processing in 2010.

Wing-Yin Yu (S'21) received the B.Eng. degree in Information Engineering from City University of Hong Kong, in 2019. He is currently pursuing the Ph.D. degree at Department of Electronic Engineering at City University of Hong Kong. His research interests are deep learning and computer vision.

Yasar Abbas Ur Rehman (S'19–M'20) received the B.Sc. degree in electrical engineering (telecommunication) from the City University of Science and Information Technology, Peshawar, Pakistan, in 2012, the M.Sc. degree in electrical engineering from the National University of Computer and Emerging Sciences, Pakistan, in 2015, and PhD degree in Electronic Engineering from City University of Hong Kong, Hong Kong, in 2019. He is currently working with TCL corporate research (HK) Co., Ltd as postdoctoral researcher. His research interests include the computer vision, machine learning, deep learning and its applications in facial recognition, biometric anti-spoofing, and video understanding.



Mengyang Liu received the B.E. degree in optoelectronic engineering from the Shanghai University of Electric Power, Shanghai, China, in 2014, and the M.Sc. degree in electronic and information engineering and the Ph.D. degree from the City University of Hong Kong, in 2015 and 2019, respectively. He is currently an Engineer with the Tencent Video, Tencent Holdings Ltd. His research interests include image and video processing, video embedding and retrieval, computer vision, and machine learning.

Yujia Zhang received the B.E. degree in electrical engineering and automation in Huazhong University of Science and Technology in 2015, and the M.S. degree in electrical engineering in South China University of Technology, China, in 2018. He is currently pursuing the Ph.D. degree in City University of Hong Kong. His current research interests include computer vision, video understanding.

Weifeng Ou received his B.Eng. degree from Guangdong University of Technology in 2013, his M.Eng. degree from South China University of Technology in 2016. He was an engineer in Huawei from 2016 to 2018. He is currently pursuing his Ph.D. degree in City University of Hong Kong. His research interests include deep learning and computer vision.