Abstract—We propose a Video Colorization with Hybrid Generative Adversarial Network (VCGAN), an improved approach to video colorization using end-to-end learning and recurrent architecture. The VCGAN addresses two prevalent issues in the video colorization domain: Temporal consistency and the unification of colorization network and refinement network into a single architecture. To enhance colorization quality and spatiotemporal consistency, the mainstream of the 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 serves 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 colorize single images. To improve the color consistency of far frames, we propose a dense long-term loss that minimizes the temporal disparity of every two remote frames. Trained with colorization and temporal losses jointly, VCGAN strikes a good balance between video color vividness and spatiotemporal continuity. Experimental results demonstrate that VCGAN produces higher-quality and temporally more consistent colorful videos than existing approaches.

Index Terms—Generative adversarial networks, placeholder feature extractor, video colorization.

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 images, 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.

Existing 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 one hand, earlier video colorization methods are often based on exemplars such as color scribbles and strokes [23]–[25]. On the other hand, to alleviate the effort of selecting proper examples, task-independent video colorization methods [27], [28] post-process framewise colorization results by adding temporal coherence. For instance, Lai et al. [28] utilized a temporal smoothing network to minimize the color differences between the two consecutive frames that are colorized individually using image colorization algorithms. However, the performance of these methods is limited by the image colorization algorithms. Furthermore, fully-automatic methods [14]–[16] learn the mapping from continuous grayscale frames to colorful frames. The mapping is normally implemented by a neural network, e.g., 3D-CNN [15], two-step network [14]. On one hand, 3D-CNN requires a large memory footprint for a long sequence (e.g., each segment is independently colorized due to the GPU memory limit). On the other hand, two-step network misses the first frame at inference, in addition to requiring large memory footprints. Moreover, the balance between color vividness and video continuity becomes another issue.

In order to address these issues, we propose to combine both image and video colorization into a hybrid architecture VCGAN. There are three main benefits of the hybrid model: 1) One model can be applied to both image and video colorization; 2) It provides reference colorized frame as a prior for the colorization of the following grayscale frames, which avoids the requirement of temporal refinement network; 3) The proposed hybrid architecture strikes a good balance between color vividness and video continuity. Firstly, we assume the continuous frames satisfy the
There are three classes of video col-


directly adopted VGG-16 as backbone

W firstly utilized a deep

Then, at the second stage, it is optimized with both colorization

VCGAN. The first stage provides good initialization weights,

we first use the large-scale image dataset ImageNet [30] to train

of frames in the video datasets are much less than image datasets,

schedule for VCGAN including single image and video col-

we adopt a patch-based discriminator for adversarial learning.

concatenated to the mainstream encoder, which are then jointly

fed into decoder. Therefore, the VCGAN generator can utilize

orized frame for video colorization. If changing the input as

approximation two neighbouring frames for video colorization. It enhances

spatiotemporal continuity of output frames. It uses the same

architecture as global feature extractor but receives the last col-

orization for objects [2], [4], [5], [29]. The second feature extractor makes VCGAN relate ev-

ery two neighbouring frames for video colorization. It enhances

the spatiotemporal continuity of output frames. It uses the same

architecture as global feature extractor but receives the last col-

orized frame for video colorization. If changing the input as

grayscale frame of current time, the VCGAN becomes an image

colorization model. The output features of both extractors are

concatenated to the mainstream encoder, which are then jointly

fed into decoder. Therefore, the VCGAN generator can utilize

these information to learn a good video colorization. In addition,

we adopt a patch-based discriminator for adversarial learning.

Regarding the optimization, we define a two-stage training

schedule for VCGAN including single image and video col-

orization, respectively. Since the total numbers and the diversity

of frames in the video datasets are much less than image datasets,

we first use the large-scale image dataset ImageNet [30] 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 [31] and Videvo [32]. In addition, we improve

the temporal smoothness of colorized frames by enforcing an ad-

ditional dense long-term loss at the second stage. It models the

dense connections of every remote frame, which is beneficial for

VCGAN to maintain the color continuity for distant frames. The

adversarial training is used to enhance the color vividness. 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 col-

orizations than the well-known methods. Some results produced

by VCGAN are shown in Fig. 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;

2) A dense long-term loss is proposed to minimize the flick-

ing artifacts of generated frames;

3) Comprehensive experiments are conducted to evaluate the

VCGAN architecture on both single image and video col-

orization 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 col-
orization methods: exemplar-based and fully-automatic. The

exemplar-based methods are based on additional user-given in-
formation such as color scribbles [23]–[25], [29] and example

colorful images [26], [33]–[35]. 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. [33] 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 fea-

tures. 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 [36] or ResNet [37]. 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 net-

works 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 tech-
nique. 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 col-
orization algorithms: exemplar-guided [17]–[19], [23]–[26],
task-independent [27], [28] and fully-automatic [14]–[16]. The

Markov chain and its transition function is implied in the model. Also, a video can include many same frames. Therefore, it is

possible to combine both image and video colorization into the

same architecture. Secondly, since the VCGAN has the ability
to process a single image (i.e., the first frame of video), there

are no frames lost compared with [14]. Thirdly, we propose a
two-stage training schedule and use the adversarial training [21]
with a dense long-term loss. They help VCGAN achieve good

colorization quality and maintain spatiotemporal continuity.

The proposed VCGAN generator architecture includes a

mainstream encoder, a mainstream decoder, and two feature ex-

tractors. The first feature extractor extracts the semantics of the

input grayscale frame, which provides high-level information

for the network to better learn colors for objects [2], [4], [5],

[29]. The second feature extractor makes VCGAN relate ev-

ey two neighbouring frames for video colorization. It enhances

the spatiotemporal continuity of output frames. It uses the same

architecture as global feature extractor but receives the last col-
orized frame for video colorization. If changing the input as

grayscale frame of current time, the VCGAN becomes an image

colorization model. The output features of both extractors are

concatenated to the mainstream encoder, which are then jointly

fed into decoder. Therefore, the VCGAN generator can utilize

these information to learn a good video colorization. In addition,

we adopt a patch-based discriminator for adversarial learning.

Regarding the optimization, we define a two-stage training

schedule for VCGAN including single image and video col-
orization, respectively. Since the total numbers and the diversity

of frames in the video datasets are much less than image datasets,

we first use the large-scale image dataset ImageNet [30] 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 [31] and Videvo [32]. In addition, we improve

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
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 the reference image and input frames to guide colorization. Jampangi 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], [38]. 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. [38] 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 $X = \{x_1, x_2, \ldots, x_T\}$. The corresponding results processed by image colorization algorithms can be represented as $Y = \{y_1, y_2, \ldots, y_T\}$ and the ground truth colorful video frames are $Z = \{z_1, z_2, \ldots, z_T\}$. 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 the previous 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 $X$ to $Z$. Therefore, the learning process of the proposed VCGAN is represented as:

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

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. Fig. 2 illustrates the different connection types of the two aforementioned representative video colorization approaches and VCGAN.

#### B. Two-Stage Training Schedule

In order to ensure that VCGAN produces perceptually plausible colorizations, the training process is divided into two stages.
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).

Fig. 3. Illustration of the training schedules for (a) the first stage and (b) the second stage. Illustration of the architectures of (c) the VCGAN generator and (d) the VCGAN discriminator, where the numbers of channels and notations of layers/blocks/operations are attached. More detailed architectures can be found in supplementary materials (https://github.com/zhaoyuzhi/VCGAN).

At the first training stage, the VCGAN performs single image colorization, as shown in Fig. 3(a). The large ImageNet dataset [30] is utilized for training since it contains much more diverse modes and categories compared with common video datasets [31], [32]. 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 colorization, the VCGAN learns \( p(z_1|x_1) \) (see (2)). For the following frames (e.g., time \( t \)), we consider the relations between every two neighbouring frames, i.e., VCGAN learns \( p(z_t|x_t, z_{t-1}) \) (see (2)). The output of time \( t-1 \) is first converted to grayscale and warped using forward flow from time \( t-1 \) to \( t \). Then, the warped image replaces the grayscale input at time \( t \) for the placeholder feature extractor, as shown in Fig. 3(b). The processes can be defined as:

\[
\begin{align*}
    s_t &= \begin{cases} 
    G(x_1), & t = 1, \\
    G(x_t, i_t), & t > 1.
    \end{cases} \\
    i_t &= W(O_{t-1 \rightarrow t}, G(p_{t-1})),
\end{align*}
\]  

(3)

(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 \( O_{t-1 \rightarrow t} \), and the operator \( G(*) \) converts RGB images to grayscale by a linear transformation. Note that \( W(*) \) and \( G(*) \) are fixed; therefore \( i_t \) is proportional to \( s_{t-1} \). Therefore, VCGAN can utilize the
information from last output, which satisfies the representation of (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 neighboring frames.

C. VCGAN Architecture

The hierarchical VCGAN generator consists of three main parts: global feature extractor, placeholder feature extractor, and mainstream encoder-decoder, as shown in Fig. 3(c). The mainstream adopts U-Net structure [39] that executes skip connection between each encoder layer and decoder layer 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 [40] 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-IN network [37] architecture, both of which are pre-trained on ImageNet [30]. 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 the two feature 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, as shown in Fig. 3(d). It contains fewer parameters than the original 1 \( \times \) 1 PixelGAN yet enhances the perceptual quality of generated samples. It also encourages sharper edges and colors.

D. Loss Functions

At the first stage, VCGAN is learned to produce accurate image colorization. The loss function of the first stage is:

\[
L_{\text{1st}} = \lambda_1 L_1 + \lambda_p L_p,
\]

where \( L_1 \) and \( L_p \) denote pixel-level reconstruction loss and perceptual loss [20], respectively. \( \lambda_1 \) and \( \lambda_p \) are trade-off coefficients. Specifically, the losses are defined as:

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

\[
L_p = \mathbb{E}[||\phi_t(s_t) - \phi_t(z)||_1],
\]

where \( s_t \) (see (3)) and \( z \) represent the colorized image and corresponding ground truth, respectively. At the first stage, \( t = 1 \). \( \phi_t(\cdot) \) produces the features from the \( t \)-th layer of a pre-trained network. In our experiment, the conv4_3 layer of VGG-16 network [36] is adopted.

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

\[
L_{\text{2nd}} = \lambda_1 L_1 + \lambda_p L_p + \lambda_G L_G + \lambda_{st} L_{st} + \lambda_{dlt} L_{dlt},
\]

where \( L_G, L_{st}, \) and \( L_{dlt} \) indicate GAN loss, short-term loss, and dense long-term loss, respectively. \( \lambda_1 \) are trade-off coefficients for each loss term.

We use the WGAN critic [41] and spectral normalization [42] in the adversarial training, which is defined as:

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

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

where (9) and (10) constitute WGAN loss for generator \( G(\cdot) \) and discriminator \( D(\cdot) \), respectively. Due to spectral normalization 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:

\[
L_{st} = \mathbb{E}\left[\sum_{t=2}^{N} M_{t-1\rightarrow t}||s_t - W(O_{t-1\rightarrow t}, s_{t-1})||_1\right],
\]

\[
L_{dlt} = \mathbb{E}\left[\sum_{t=3}^{N} \sum_{m=1}^{t-2} M_{m\rightarrow t}||s_t - W(O_{m\rightarrow t}, s_m)||_1\right],
\]

where \( N \) is the numbers of frames in a batch, \( s_m \) and \( s_t \) are the colorized frames at time \( m \) and \( t \), respectively. \( M_{m\rightarrow t} \) and \( O_{m\rightarrow t} \) represent the non-occlusion mask [28] and real forward flow of colorful images between time \( m \) and \( t \), respectively. The operator \( W(\cdot) \) warps input frame under the guidance of flow \( O_{m\rightarrow t} \). 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:

\[
M_{m\rightarrow t} = \exp(-\alpha||x_t - W(O_{m\rightarrow t}, x_m)||_2^2),
\]

where the mask \( M_{m\rightarrow t} \) indicates the non-occlusion regions of the warped image. The scale factor \( \alpha \) enlarges the numerical disparity between occlusion and non-occlusion regions.

IV. EXPERIMENT

A. Implementation Details

Dataset: We use the entire ImageNet [30] dataset (1281167 images with 1000 categories) at the first training stage. The images are resized to 256 \( \times \) 256. The images encoded as grayscale are excluded. At the second training stage, we utilize the DAVIS [31] and Video [32] datasets that contain 156 short videos (overall 29620 images). We assume each short video is equally important when selecting data for training. All training images are normalized to the range of \([-1, 1] \).

Network: Both the generator and discriminator adopt LeakyReLU [43] activation function. The instance normalization [44] 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 [37] also adopts
LeakyReLU [43] activation function and instance normalization [44]. Specifically, to maintain more information while performing the down-sampling operation, the pooling layers of the original ResNet-50-IN architecture [37], [44] 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 [37] 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 initialized with Xavier method [45].

**Optimization:** For the first stage, the generator of VCGAN is trained with (5) for 20 epochs. The learning rate is initialized to $2 \times 10^{-4}$, 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 trained with (8) on continuous frames with 256p resolution and 480p resolution, for 500 epochs and 500 epochs, respectively. The initial learning rates for both generator and discriminator equal to $5 \times 10^{-5}$. For 480p resolution, the learning rate is halved every 100 epochs. For a single category, we randomly sample $N = 5$ successive frames at one iteration. The scale factor $\alpha$ of the non-occlusion mask $M$ (see (11), (12), and (13)) is set to 50. For the optimization, we use Adam optimizer [46] with $\beta_1 = 0.5$ and $\beta_2 = 0.999$. The batch size equals to 16 and 4 for the two stages, respectively.

The coefficients $\lambda_1, \lambda_p, \lambda_{cl}, \lambda_{st}, \lambda_{dl}$ are empirically set to 10, 5, 1, 3, 5, respectively. At the first stage, the VCGAN is trained on 4 NVIDIA Titan Xp GPUs (12 Gb). 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 Python 3.6. The whole training of VCGAN takes approximately 14 days, where 10, 1, and 3 days for the first stage, second stage on 256p and 480p, respectively.

**B. Experiment Settings**

**Dataset:** Following [14], [28], we perform the evaluations on DAVIS [31] and Videvo [32] testing set. 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...
Fig. 5. Colorization comparison on “SkateboarderTableJump” from Videvo [32] dataset. The first and last rows 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], 3DVC [15], FAVC [14], and the proposed VCGAN. The red rectangles highlight inconsistent regions or strange colors for the baselines.

TABLE I

| Method                  | DAVIS PSNR | SSIM     | Warp Error | VIDEVO 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    | /              | /              | /            |
| SFCGAN [12]            | 23.19      | 0.8595   | 0.05918    | 23.29       | 0.8549   | 0.03301    | /              | /              | /            |
| CIC + BTC [28]         | 21.48      | 0.8898   | 0.05170    | 21.02       | 0.8809   | 0.02891    | /              | /              | /            |
| LTBC + BTC [28]        | 22.45      | 0.9006   | 0.05144    | 22.81       | 0.9072   | 0.02995    | /              | /              | /            |
| ChromaGAN + BTC [28]   | 19.88      | 0.8896   | 0.04955    | 16.63       | 0.8289   | 0.02753    | FlowNet2       | /              | /            |
| SFCGAN + 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       | 0.9209   | 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    | /              | /              | /            |

The red, blue, and green colors represent the best, the second-best, and the third-best performances, respectively.

...may produce images of diverse resolutions, all the result images are generated and resized to match the image resolution of ground truth for fairness. 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 use the 10000 ImageNet validation images [30] as same as [2], [4], [11], [12].

PSNR and SSIM [47]: 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 [47] 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...
TABLE II
COMPARISON ON MULTIPLY–ACCUMULATE OPERATIONS (MACS) AND GPU MEMORY PER BATCH (MPB). WE SELECT TWO IMAGE RESOLUTIONS 256×512 (256R) AND 512×1024 (512R)

| Method     | 256R MACs | 256R MPB | 512R MACs | 512R MPB |
|------------|-----------|----------|-----------|----------|
| 3DVC       | 79.72 Mb  | OM>50 T  | 318.87 Mb | OM>15 T  |
| FAVC       | 126.69 Mb | 2365 Mb  | 506.79 Mb | 7237 Mb  |
| VCGAN      | 60.52 Mb  | 1240 Mb  | 242.11 Mb | 2074 Mb  |

Fig. 6. The experiment results on memory consumption. The experiments run on a single NVIDIA Titan Xp GPU with total 12000 Mb memory. “OM” denotes “out of memory” (i.e., more than 12000 Mb).

C. Video Colorization Comparisons

We compare VCGAN with existing 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 all compared methods are the same to VCGAN (i.e., ImageNet [30], DAVIS [31], and Videvo [32]).

Qualitative comparison: Some generated samples from typical methods on two validation sets are shown in Figs. 4 and 5. On the one hand, the single image colorization methods CIC [4] and LTBC [5] produce temporally inconsistent results. As shown in the highlighted patches, the colors of objects change extremely, e.g., the sand in Fig. 4 and the arm in Fig. 5. It is because these methods do not consider inter-frame relations. It also demonstrates these image colorization methods are not very robust to shifts or motions. On the other hand, the post-processed results (i.e., CIC + BTC and LTBC + BTC) do not address the above issue obviously. Though BTC [28] induces the temporal relations between every two frames, the colors of image colorization results are “too different”. BTC cannot handle such cases; therefore, BTC’s results are still not continuous enough. In addition, BTC is not jointly trained with single image colorization methods such as CIC and LTBC; therefore, the optimization of image colorization and temporal smoothing are separated. It also causes flickering artifacts in the generated frames. For video colorization algorithms 3DVC [15], FAVC [14], and VCGAN, they do not encounter such issues since temporal correlations are modeled. However, the objects (e.g., the sky, horse, and man) in both Figs. 4 and 5 colorized by FAVC are dusky. The colors of 3DVC results are not natural enough, e.g., the man in Fig. 5.

Among all the methods, VCGAN produces more colorful and temporally coherent frames. We include more samples and video clips in supplementary materials.

Quantitative comparison: The quantitative comparison on 480p validation sets is concluded in Table I. The results of grayscale frames serve as a baseline. Note that we do not include single image colorization methods [4], [5], [11], [12] in comparisons since they do not consider the temporal continuity. Firstly,
BTC [28] 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). However, the results from CIC + BTC suffer from a decrease of PSNR compared with CIC, since the frame-wise characteristic might be weakened. As discussed, since BTC and CIC are not jointly trained, it also causes decreases in metrics. A similar conclusion also applies to LTBC, ChromaGAN, and SCGAN. Secondly, FA VC [14] uses a two-step network, which is jointly optimized by colorization loss function (e.g., L1 loss) and temporal smoothing loss function (e.g., short-term loss). It achieves better PSNR and SSIM results than post-processed results from BTC. Thirdly, 3DVC [15] learns both spatial and temporal relations by 3D Conv instead of two-step networks. Therefore, it achieves better PSNR, SSIM, and Warp Error results than FAVC. However, it still adopts a simple 3D U-Net [39] architecture, which restricts its performances. Finally, VCGAN 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 \( L_{dit} \) are obviously beneficial to video colorization. The proposed VCGAN uses the semantic model (i.e., two feature extractors), which promotes fast convergence and high colorization quality. In addition, the VCGAN

**TABLE IV**

| Setting | \( \lambda_1 \) | \( \lambda_p \) | \( \lambda_C \) | \( \lambda_L \) | \( \lambda_{dit} \) | Target | DAIS | PSNR | SSIM | Warp Error | Video | PSNR | SSIM | Warp Error |
|---------|-----------------|-----------------|-----------------|-----------------|-----------------|--------|-------|-------|---------|--------|-------|-------|---------|
| s(1)    | 1               | 1               | 1               | 1               | 1               | all “I” coefficients | 23.83 | 0.9193 | 0.05101 | 24.68  | 0.9224 | 0.02659 |
| s(2)    | 20              | 5               | 1               | 1               | 1               | 3                  | 23.90 | 0.9192 | 0.05042 | 25.11  | 0.9244 | 0.02746 |
| s(3)    | 10              | 10              | 1               | 3               | 5               | 5                  | 23.85 | 0.9202 | 0.04971 | 25.20  | 0.9232 | 0.02602 |
| s(4)    | 10              | 5               | 2               | 3               | 5               | 5                  | 23.32 | 0.9113 | 0.04957 | 24.66  | 0.9211 | 0.02644 |
| s(5)    | 10              | 5               | 1               | 6               | 5               | 6                  | 23.75 | 0.9133 | 0.04915 | 24.55  | 0.9197 | 0.02565 |
| s(6)    | 10              | 5               | 1               | 3               | 10              | 10                 | 23.37 | 0.9096 | 0.04909 | 24.67  | 0.9194 | 0.02536 |
| s(7)    | 20              | 10              | 2               | 3               | 5               | 5                  | 23.63 | 0.9118 | 0.04933 | 25.07  | 0.9245 | 0.02650 |
| s(8)    | 10              | 5               | 1               | 6               | 10              | 5                  | 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

The Red, Blue, and green Colors Represent the Best, the Second-Best, and the Third-Best Performances, Respectively.
is the only hybrid model that unifies both image and video colorization in the same architecture.

D. Memory Analysis

We conduct a memory analysis for VCGAN and two existing video colorization methods [14], [15] on two image resolutions. The results are concluded in Fig. 6 and Table II. In the experiment, we adjust the number of grayscale frames in a video (to be colorized by the colorization methods) to compute the memory consumption of different methods. The computing platform is one NVIDIA Titan Xp GPU with 12 Gb memory. It is clear that VCGAN has the minimum theoretical MACs and the smallest memory consumption among all the methods. Since 3DVC [15] uses 3D Conv, it only ensures the input sequence is temporally related. However, the relations between different sequences are not modeled. Therefore, it easily causes flickering artifacts for long videos due to memory limits. If feeding a longer sequence to 3DVC, it easily encounters “out of memory,” as shown in Fig. 6. FAQC processes two frames simultaneously; however, it has much larger theoretical MACs due to the hyper-column operation. Applying FAQC to color videos still causes much more memory consumption than VCGAN.

E. 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 [31] and Video [32] 480p validation data. There are 20 settings with abbreviations as shown in Table III.

Loss terms: The settings l(1), l(2.1), and l(2.2) 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 Fig. 7.
Fig. 9. The comparison of smoothing ability on “cows” from DAVIS [31] dataset.

Fig. 10. The comparison of the utility of the proposed dense long-term loss $L_{dlt}$ on “Ducks” from Videvo [32] dataset.

**Loss terms related to colorization quality:** As shown in Fig. 7(a), the baseline setting l(1) obtains the worst result since it only uses $L_1$ for training. For the settings l(3.1)-l(3.3), if VCGAN trained without $L_1$, $L_p$, or $L_G$, there is a drop in terms of PSNR and SSIM metrics, which demonstrate that all of them are beneficial for colorization quality. As shown in Fig. 8, $L_p$ or $L_G$ promotes VCGAN to generate more realistic and vivid colorizations. Similarly, for the settings l(3.4)-l(3.7), their results are worse than full loss terms, since more than one “colorization quality”-related loss is 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 Fig. 8, 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:** As shown in Fig. 7(b), setting l(1) obtains the worst Warp Error. By adding the short-term loss $L_{st}$ (setting l(2.1)), VCGAN has better results since it meets the Markov Chain assumption. By adding the dense long-term loss $L_{dlt}$ (setting l(2.2)), VCGAN also performs better due to the consideration of temporal relations. 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 and 0.00246 increases on DAVIS and Videvo, respectively). Similarly, the Warp Error increases if removing $L_{ldt}$ (i.e., l(4.2)) or both $L_{st}$ and $L_{ldt}$ (i.e., l(4.3)). As shown in Fig. 9, 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 $L_{ldt}$ 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 $L_{ldt}$:** Some previous methods set the time range equals to 2 (previous and current frame) [14], [15], [18], [25], [49] or 3 (previous, current, and leading frames) [48], [50]. 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 $L_{ldt}$ 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 $L_{ldt}$, i.e., l(4.2), l(4.4), and l(4.1). In terms of Warp Error, $L_{st}$ (l(4.1)) is the most significant factor to smooth videos since it panels the neighbouring frames. Though $L_{ldt}$ (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 $L_{ldt}$ (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 $L_{ldt}$ with $L_{lt}$. As shown in Fig. 7(b) full VCGAN setting, $L_{ldt}$ reduces Warp Errors by approximately 0.00500 and 0.00338 on DAVIS and Video datasets, respectively, than $L_{lt}$. In addition, since the continuous frames may not represent long-term consistency, we illustrate remote frames in Fig. 10 to show the effect of the proposed dense long-term loss $L_{ldt}$. Only the VCGAN trained
with $L_{	ext{diff}}$ produces consistent background color (i.e., blue sky); whereas the normal long-term loss $L_{	ext{lt}}$ 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 Fig. 11, the patches are less colorful 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. Some results are shown in Fig. 11, where the colors are not vivid enough and the frames are not continuous enough compared with the 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.

**F. Investigation of the Sensitiveness of Loss Coefficients**

The coefficients of the objectives used for VCGAN optimization are empirically selected. To demonstrate that the proposed coefficient combination is relatively better than other combinations, we conduct several experiments by adjusting some of the coefficients. The results on DAVIS and Video 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 Fig. 12, if VCGAN trained with all “1” coefficients (i.e., s(1)), the colors are very consistent for far frames, since it may not balance high-quality colorization and temporal consistency well. If doubling $\lambda_{G}$ (i.e., s(4)), the results are almost monochrome. If doubling $\lambda_{sl}$ (i.e., s(5)), the colors are also less vivid. In conclusion, the proposed coefficient combination is relatively better.

**G. 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], [51], where the colorization part of [14] is adopted. Note that, the training sets of the methods are the same (i.e., ImageNet [30]). Following the settings in [4], we choose the 10000 images from the ImageNet validation set for evaluation.

We illustrate some colorized results in Fig. 13. There are obvious visual artifacts in the generated results of other methods. For instance, there are color bleeding artifacts (i.e., the color of one object permeates to other objects) in rows 1-4 of CIC [4] and rows 3 and 4 of DeOldify [51]. Even though there are no color bleeding artifacts of PAVC [14], their 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 of VCGAN. 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 evaluate 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 [51], ChromaGAN [11] and the proposed VCGAN) obtain better performance, since the GAN facilitates sharper results, which are difficult to accomplish by only adopting L1 loss.

**H. Failure Cases**

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 (please refer to the left part of Fig. 14). Also, since the video colorization is an ill-posed problem, the produced frames might be not colorful.
Fig. 13. Illustration of image colorization results of VCGAN (first stage) and state-of-the-art methods [3]–[5], [11], [12], [14], [51] 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.

Fig. 14. Failure cases of VCGAN. The rows from top to bottom denote the grayscale input, colorized frames by VCGAN, and ground truth, respectively.

enough (please refer to the right part of Fig. 14). 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 to automatically generate photorealistic and temporally coherent video colorization. Utilizing two pre-trained ResNet-50-IN networks as the global feature extractor and placeholder feature extractor along with the U-Net-based mainstream, the VCGAN generator 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 every remote relations for far frames. It enhances the smoothness of generated videos while requires ignorable additional memory. The adversarial loss is also adopted in the video colorization domain to improve the color vividness. Finally, we validated VCGAN with several state-of-the-art image and video colorization methods. The experiment shows that VCGAN has the minimum theoretical MACs and the smallest memory consumption among current video colorization methods. The experiment also demonstrates that the proposed VCGAN obtains better performances.
in both image and video colorization applications than the other well-known methods.

ACKNOWLEDGMENTS

The authors would like to thank Bei Li, Pengfei Xian, Xuhiui 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. IEEE Int. Conf. Comput. Vis., 2015, pp. 415–423.
[2] G. Liessson, M. Maire, and G. Shakhnarovich, “Learning representations for automatic colorization,” in Proc. Eur. Conf. Comput. Vis., 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. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 1125–1134.
[4] R. Zhang, P. Isola, and A. A. Efros, “Colorful image colorization,” in Proc. Eur. Conf. Comput. Vis., 2016, pp. 649–666.
[5] S. Iizuka, E. Simo-Serra, and H. Ishikawa, “Learning diverse image colorization,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 6837–6845.
[6] N. H. Thasarathan, K. Nazeri, and M. Ebrahimi, “Automatic temporally co-

[7] [14] C. Lei and Q. Chen, “Fully automatic video colorization with self-

[8] [21] I. Goodfellow et al., “Generative adversarial networks,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 5792–5801.
[9] [32] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein generative adversarial networks,” in Proc. Int. Conf. Learn. Representations, 2017, pp. 1–15.
[10] [34] Y.-W. Tai, J. Jia, and C.-K. Tang, “Local color transfer via probabilistic segmentations by expectation-maximization,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 770–778.
[11] [36] X. Liu et al., “Intrinsic colorization,” ACM Trans. Graph., vol. 27, no. 5, pp. 1–9, 2008.
[12] [37] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in Proc. Int. Conf. Learn. Representations, 2015, pp. 1–14. [Online]. Available: https://arxiv.org/abs/1409.1556
[13] [42] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, “Spectral normalization for generative adversarial networks,” in Proc. Int. Conf. Learn. Representations, 2018, pp. 1–12. [Online]. Available: https://openreview.net/forum?id=B1QRgziT-

[14] [44] D. Ulyanov, A. Vedaldi, and V. Lempitsky, “Instance normalization: The missing ingredient for fast stylization,” 2016, arXiv preprint arXiv:1607.08022.
[15] [48] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein generative adversarial networks,” in Proc. Int. Conf. Learn. Representations, 2017, pp. 1–15. [Online]. Available: https://openreview.net/forum?id=8gmWwjFyLj
[16] [50] R. Y. Zhang et al., “Real-time user-guided image colorization with learned deep priors,” ACM Trans. Graph., vol. 36, no. 4, 2017, Art. no. 119.
[17] [53] T. Miyato, T.-C. Wang et al., “Coherent, volumetric GAN for super-resolution fluid flow,” in Proc. NeurIPS, 2019, pp. 7576–7586.
[18] [55] T.-C. Wang, T.-Y. Lin, and J. Xiao, “End-to-end learning of global and local image priors for automatic image colorization with simultaneous classification,” ACM Trans. Graph., vol. 35, no. 4, pp. 1–11, 2016.
[19] [57] T. Miyato, T.-C. Wang et al., “Coherent, volumetric GAN for super-resolution fluid flow,” in Proc. NeurIPS, 2019, pp. 7576–7586.
[20] [61] T. Ghiasi, M. Ashikhmin, and K. Mueller, “Transfering color to greyscale images,” in Proc. Int. Conf. Comput. Graph. Interactive Techn., 2002, pp. 277–280.
[21] [62] Y.-W. Tai, J. Jia, and C.-K. Tang, “Local color transfer via probabilistic segmentations by expectation-maximization,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 724–732.
[22] [63] Videvo, Accessed Sep. 30, 2018. [Online]. Available: https://www.videvo.org
[23] [65] W.-S. Lai et al., “Learning blind video temporal consistency,” in Proc. Eur. Conf. Comput. Vis., 2018, pp. 170–185.
[24] [66] X. Liu et al., “Intrinsic colorization,” ACM Trans. Graph., vol. 27, no. 5, pp. 1–9, 2008.
[25] [67] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in Proc. Int. Conf. Learn. Representations, 2015, pp. 1–14. [Online]. Available: https://arxiv.org/abs/1409.1556
[26] [68] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 770–778.
[27] [69] J.-Y. Zhu et al., “Toward multimodal image-to-image translation,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2017, pp. 465–476.
[28] [70] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional networks for biomedical image segmentation,” in Proc. Int. Conf. Med. Image Comput. Comput.-Assisted Intervention, 2015, pp. 234–241.
[29] [71] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, “Self-attention generative adversarial networks,” Proc. Int. Conf. Mach. Learn., pp. 7354–7363, 2019, arXiv:1805.08318.
[30] [72] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein generative adversarial networks,” in Proc. 34th Int. Conf. Mach. Learn., 2017, pp. 214–223.
[31] [73] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, “Spectral normalization for generative adversarial networks,” in Proc. Int. Conf. Learn. Representations, 2018, pp. 1–26. [Online]. Available: https://openreview.net/forum?id=B1QRgziT-

[32] [74] A. L. Maas, A. Y. Hannun, and A. Y. Ng, “Rectifier nonlinearities improve neural network acoustic models,” in Proc. Int. Conf. Mach. Learn., 2013, Art. no. 3.
[33] [75] D. Ulyanov, A. Vedaldi, and V. Lempitsky, “Instance normalization: The missing ingredient for fast stylization,” 2016, arXiv preprint arXiv:1607.08022.
[34] [76] X. Ghorbani and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” in Proc. 13th Int. Conf. Artif. Intell. Stat., 2010, pp. 249–256.
[35] [77] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in Proc. Int. Conf. Learn. Representations, 2015, pp. 1–15. [Online]. Available: https://openreview.net/forum?id=8gmWwjFyLj
[36] [78] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: From error visibility to structural similarity,” IEEE Trans. Image Process., vol. 13, no. 4, pp. 600–612, Apr. 2004.
[37] [79] M. Ruder, A. Dosovitskiy, and T. Brox, “Artistic style transfer for videos,” in Proc. German Conf. Pattern Recognit., 2016, pp. 26–36.
[38] [80] Y. Xie, E. Franz, M. Chu, and N. Thurey, “tempoGAN: A temporally coherent, volumetric GAN for super-resolution fluid flow,” ACM Trans. Graph., vol. 37, no. 4, 2018, Art. no. 95.
[39] [81] D. Kim, S. Woo, J.-Y. Lee, and I. S. Kweon, “Deep video inpainting,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 5792–5801.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
Yuzhi Zhao (Graduate Student Member, IEEE) received the B.Eng. degree in electronic information from the Huazhong University of Science and Technology, Wuhan, China, in 2018. He is currently working toward the Ph.D. degree with the Department of Electronic Engineering, City University of Hong Kong, Hong Kong. His research interests include image processing, deep learning, and machine learning.

Lai-Man Po (Senior Member, IEEE) 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. Since 1991, he has been with the Department of Electronic Engineering, City University of Hong Kong, where he is currently an Associate Professor with the Department of Electronic Engineering. He has authored more than 150 technical journal and conference papers. His research interests include image and video coding with an emphasis deep learning based computer vision algorithms. He 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. From 2011 to 2013, he was an Associate Editor for the HKIE Transactions. He was also on the Organizing Committee of the IEEE International Conference on Acoustics, Speech and Signal Processing in 2003, and IEEE International Conference on Image Processing in 2010.

Wing-Yin Yu (Graduate Student Member, IEEE) received the B.Eng. degree in information engineering from the City University of Hong Kong, Hong Kong, in 2019. He is currently working toward the Ph.D. degree with the Department of Electrical Engineering. His research interests include deep learning and computer vision.

Yasar Abbas Ur Rehman (Member, IEEE) 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, Islamabad, Pakistan, in 2015, and the Ph.D. degree in electronic engineering from the City University of Hong Kong, Hong Kong, in 2019. He is currently a Postdoctoral Researcher with TCL Corporate Research Co., Ltd, Hong Kong. His research interests include computer vision, machine learning, deep learning and its applications in facial recognition, biometric anti-spoofing, and video understanding.

Mengyang Liu received the B.Eng. 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, 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 (Graduate Student Member, IEEE) received the B.Eng. degree in electrical engineering and automation from the Huazhong University of Science and Technology, Wuhan, China, in 2015, and the M.Sc. degree in electrical engineering from the South China University of Technology, Guangzhou, China, in 2018. He is currently working toward the Ph.D. degree with the City University of Hong Kong, Hong Kong. His research interests include computer vision and video understanding.

Weifeng Ou received the B.Eng. degree in telecommunications engineering from the Guangdong University of Technology, Guangzhou, China, in 2013, the M.Eng. degree in signal and information processing from the South China University of Technology, Guangzhou, China, in 2016, and the Ph.D. degree from the Department of Electrical Engineering, City University of Hong Kong, Hong Kong, in 2021. From 2016 to 2018, he was an R&D Engineer with Huawei. He is currently with SenseTime Group Limited. His research interests include biometrics and deep learning.