Deep Exemplar-based Video Colorization

Bo Zhang[1], Mingming He[1,2], Jing Liao[3], Pedro V. Sander[1],
Lu Yuan[4], Amine Bermak[1], and Dong Chen[5]

[1]Hong Kong UST
[2]USC Institute for Creative Technologies
[3]City University of Hong Kong
[4]Microsoft AI Perception and Mixed Reality
[5]Microsoft Research

Abstract

This paper presents the first end-to-end network for exemplar-based video colorization. The main challenge is to achieve temporal consistency while remaining faithful to the reference style. To address this issue, we introduce a recurrent framework that unifies the semantic correspondence and color propagation steps. Both steps allow a provided reference image to guide the colorization of every frame, thus reduce accumulated propagation errors. Video frames are colorized in sequence based on the history of colorization, and its coherency is further enforced by the temporal consistency loss. All of these components, learnt end-to-end, help produce realistic videos with good temporal stability. Experiments show our result is superior to the state-of-the-art methods both quantitatively and qualitatively.

1. Introduction

Prior to the advent of automatic colorization algorithms, artists revived legacy images or videos through a careful manual process. Early image colorization methods relied on user-guided scribbles [1, 2, 3, 4, 5] or a sample reference [6, 7, 8, 9, 10, 11, 12, 13] to address this ill-posed problem, and more recent deep-learning works [14, 15, 16, 17, 18, 19, 20] directly predict colors by learning color-semantic relationships from a large database.

A more challenging task is to colorize legacy videos. Independently applying image colorization (*e.g.*, [15, 16, 17]) on each frame often leads to flickering and false discontinuities. Therefore there have been some attempts to impose temporal constraints on video colorization. A naïve approach is to run a temporal filter on the per-frame colorization results as a post-processing [21, 22], which can alleviate the flickering but cause color fading and blurring. Another kind of approaches propagate the color scribbles from one frame to the following according to optical flow [1, 2, 23, 24, 25]. However, scribbles propagation may be not perfect due to flow error, which will induce some visual artifacts. The most recent methods assume that the first frame is colorized and then propagate its colors to the following frames [26, 27, 28, 29]. This is effective to colorize a short video clip, but the errors will progressively accumulate when the video is long. These existing techniques are generally based on color propagation and do not consider the content of all frames when determining the colors.

We instead propose a method to colorize video frames jointly considering three aspects, instead of solely relying on the previous frame. First, our method takes the result of the previous frame as input to preserve temporal consistency. Second, our method performs colorization using an exemplar, allowing a provided reference image to guide the colorization of every frame, thus reduce accumulated errors. Thus, finding semantic correspondence between the reference and every frame is essential to our method. Finally, our method leverages large-scale data for learning, so that it can predict natural colors based on the semantics of the input grayscale image when no proper matching is available in either the reference image or the previous frame.

To achieve the above objectives, we present the first end-to-end convolutional network for exemplar-based video colorization. It is a recurrent structure that allows history information passing to the present for keeing consistency. Each state consists of two major modules: a correspondence subnet to align the reference to the input frame based on dense semantic correspondences, and a colorization subnet to colorize a frame guided by both the colorized result of its previous frame and the aligned reference. All subnets are jointly

trained, yielding multiple benefits. First, the jointly trained correspondence subnet is tailored for the colorization task, thus achieving higher quality. Second, it is two orders of magnitude faster than the state-of-the-art exemplar-based colorization method [30] where the reference is aligned in a pre-processing step using a slow iterative optimization algorithm [31]. Moreover, the joint training allows adding temporal constraints on the alignment as well, which is essential to consistent video colorization. This entire network is trained in an unsupervised way with novel loss functions considering natural occurrence of colors, faithfulness to the reference, spatial smoothness and temporal coherence.

The experiments demonstrate that our video colorization network outperforms existing methods quantitatively and qualitatively. Moreover, our video colorization allows two modes. If the reference is a colorized frame in the video, our network will perform the same function as previous color propagation methods but in a more robust way. More importantly, our network supports colorizing a video with a color reference of a different scene. This allows the user to achieve customizable multimodal results by simply feeding various references, which cannot be accomplished in previous video colorization methods.

2. Related work

Interactive Colorization. Early colorization methods focus on using local user hints in the form of color points or strokes [1, 2, 3, 4, 5]. The local color hints are propagated to the entire image according to the assumption that coherent neighborhoods should have similar colors. These pioneering works rely on the hand-crafted low-level features for the color propagation. Recently, Zhang and Zhu et al. [32] proposed to employ deep neural networks to propagate the user edits by incorporating semantic information and achieve remarkable quality. However, all of these user-guided methods require significant manual interactions and aesthetic skills to generate plausible colorful images, making them unsuitable for colorizing images massively.

Exemplar-based Colorization. Another category of work colorize the grayscale images by transferring the color from the reference image in a similar content. The pioneering work [6] transfers the chromatic information to the corresponding regions by matching the luminance and texture. In order to achieve a more accurate local transfer, various correspondence techniques have been proposed by matching low-level hand-crafted features [7, 8, 9, 10, 11, 12, 13]. Still, these correspondence methods are not robust to complex appearance variations of the same object because low-level features do not capture semantic information. More recent works [33, 30] rely on the Deep Analogy method [31] to establish the semantic correspondence and then refine the colorization by solving Markov random field model [33] or a neural network [30]. In those work, the correspondence and the color propagation are optimized independently, therefore visual artifacts tend to arise due to correspondence error. On the contrary, we unify the two stages within one network, which is trained end-to-end and produces more coherent colorization results.

Fully Automatic Colorization. With the advent of deep learning techniques, various fully automatic colorization methods have been proposed to learn a parametric mapping from grayscale to color using large datasets [14, 15, 16, 17, 18, 19, 20]. These methods predict the color by incorporating the low and high-level cues and have shown compelling results. However, these methods lack the modelling of color ambiguity and thus cannot generate multimodal results. In order to address these issues, diverse colorization methods have been proposed using the generative models [34, 35, 36, 37, 38]. However, all of these automatic methods are prone to produce visual artifacts such as color bleeding and color washout, and the quality may significantly deteriorate when colorizing objects out of the scope of the training data.

Video Colorization. Comparatively, much less research efforts focused on video colorization. Existing video colorization can be classified into three categories. One is to post-process the frame-wise colorization with general temporal filter [21, 22], but these works tend to wash out the colors. Another class of methods propagate the color scribbles to other frames by explicitly calculating the optical flow [1, 2, 23, 24, 25]. However, scribbles drawn from one specific image may not be suitable for other frames. Another category of video colorization methods use one colored frame as an example and colorize the following frames in sequence. While conventional methods rely on hand-crafted low-level features to find the temporal correspondence [39, 40, 41], a recent trend is to use a deep neural network to learn the temporal propagation in a data-driven manner [26, 27, 28, 29]. These approaches generally achieve better quality. However, a common issue of these video color propagation methods is that the color propagation will be problematic if it fails on a particular frame. Moreover, these methods require a good colored frame to bootstrap, which can be challenging in some scenes, particularly when it is dynamic and with significant variations. By contrast, our work refers to an example reference image during the entire process, thus not relying solely on color propagation from previous frames. It therefore yields more robust results, particularly for longer video clips.

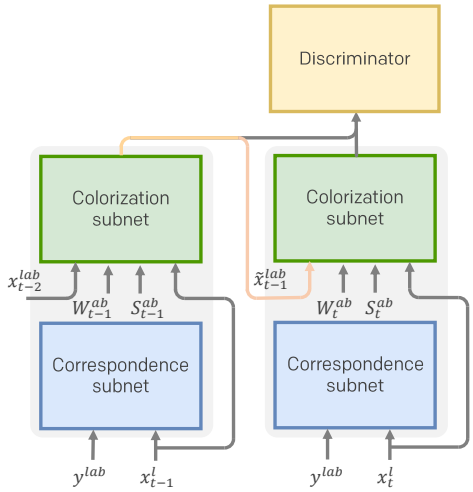

Figure 1. The framework of our video colorization network. The network consists of two subnets: correspondence subnet and colorization subnet. The colorization for the frame x_t^l is conditional on the previous colorized frame x_{t-1}^l

.

3. Method

3.1. Overall framework

We denote the grayscale video frame at time t as $x_t^l \in \mathbb{R}^{H\times W\times 1}$, and the reference image as $y^{lab} \in \mathbb{R}^{H\times W\times 3}$. Here, l and ab represent the luminance and chrominance in LAB color space, respectively. In order to generate temporally consistent videos, we let the network, denoted by $\mathcal{G}_V$, colorize video frames based on the history. Formally, we formulate the colorization for the frame $\tilde{x}_t^l$ to be conditional on both the colorized last frame $\tilde{x}_{t-1}^{lab}$ and the reference y^{lab}:

$$\tilde{x}_t^{ab} = \mathcal{G}_V(x_t^l|\tilde{x}_{t-1}^{lab}, y^{lab}) \tag{1}$$

The pipeline for video colorization is shown in Figure 1. We propose a two-stage network which consists of two subnets - correspondence network $\mathcal{N}$ and colorization network $\mathcal{C}$. At time t, first $\mathcal{N}$ aligns the reference color y^{ab} to x_t^l based on their semantic correspondences, and yields two intermediate outputs: the warped color W^{ab} and a confidence map S measuring the correspondence reliability. Then $\mathcal{C}$ uses the warped intermediate results along with the colorized last frame $\tilde{x}_{t-1}^{lab}$ to colorize $\tilde{x}_t^l$. Thus, the network colorizes the video frames in sequence and Eq. 1 can be expressed as:

$$\tilde{x}_t^{ab} = \mathcal{C}(x_t^l, \mathcal{N}(x_t^l, y^{lab})|\tilde{x}_{t-1}^{lab}) \tag{2}$$

3.2. Network architecture

Figure 2 illustrates the two-stage network architecture. Next we describe these two sub networks.

Correspondence Subnet. We build the semantic corresondence between x_t^l and y using the deep features extracted from the VGG19 [42] pretrained on imge classification. In $\mathcal{N}$, we extract the feature maps from layers of $relu2\_2$, $relu3\_2$, $relu4\_2$ and $relu5\_2$ for both x^l and y. The multi-layer feature maps are concatenated to form features $\Phi_x, \Phi_y \in \mathbb{R}^{H\times W\times C}$ for x_t^l, y respectively. Features Φ_x and Φ_y are fed into several residual blocks to better exploit the features from different layers, and the outputs are reshaped into two feature vectors: $F_x, F_y \in \mathbb{R}^{HW\times C}$ for x_t^l and y respectively. The residual blocks, parameterized by $\theta_\mathcal{N}$, share the same weights for x_t^l and y.

Given the feature representation, we can find dense correspondence by calculating the pairwise similarity between the features of x_t^l and y. Formally, we compute a correlation matrix $\mathcal{M} \in \mathbb{R}^{HW\times HW}$ whose elements characterize the similarity of F_x at position i and F_y at j:

$$\mathcal{M}(i,j) = \frac{(F_x(i) - \mu_{F_x}) \cdot (F_y(j) - \mu_{F_y})}{\|F_x(i) - \mu_{F_x}\|_2 \, \|F_y(j) - \mu_{F_y}\|_2} \tag{3}$$

where μ_{F_x} and μ_{F_y} represent mean feature vectors. We empirically find such normalization makes the learning more stable. Then we can warp the reference color y^{ab} towards x_t^l according to the correlation matrix. We propose to calculate the weighted sum of y^{ab} to approximate the color sampling from y^{ab}:

$$\mathcal{W}^{ab}(i) = \sum_j \underset{j}{\text{softmax}}(\mathcal{M}(i,j)/\tau) \cdot y^{ab}(i,j) \tag{4}$$

We set $\tau = 0.01$ so that the row vector $\mathcal{M}(i,\cdot)$ approaches to one-hot vector and weighted color $\mathcal{W}^{ab}$ approximates selecting one pixel in the reference with largest similarity score. The resulting vector $\mathcal{W}^{ab}$ serves as an aligned color reference to guide the colorization in the next step. Note that Equation 4 has a close relationship with the non-local operator proposed by Wang et al. [43]. The major difference is that the non-local operator computes the pairwise similarity within the same feature map so as to incorporate global information, whereas we compute the pairwise similarity between features of different images and use it to warp the corresponding color from the reference.

Given that the color warping is not accurate everywhere, we output the matching confidence map S indicating the reliability of sampling the reference color for each position i of x_t^l:

$$\mathcal{S}(i) = \max_j \mathcal{M}(i,j) \tag{5}$$

In summary, our correspondence network generates two outputs: warped color $\mathcal{W}^{ab}$ and confidence map S:

$$(\mathcal{W}^{ab}, \mathcal{S}) = \mathcal{N}(x_t^l, y^{lab}; \theta_\mathcal{N}) \tag{6}$$

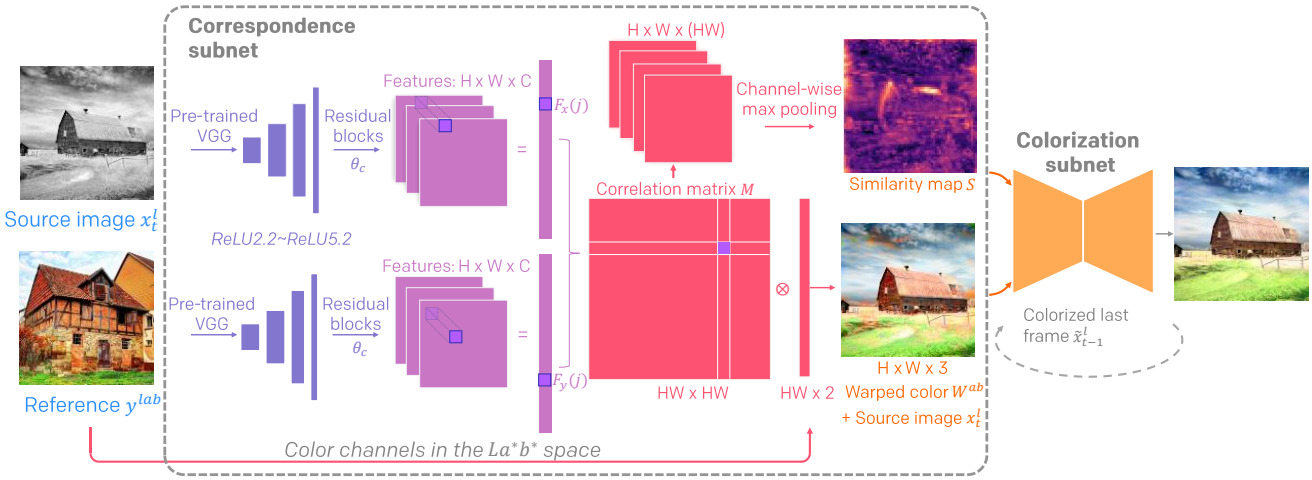

Figure 2. The detailed diagram of the proposed network. The correspondence subnet finds the correspondence of of source image x_t^l and reference image y^{lab} in the deep feature domain, and aligns the reference color accordingly. Based on the intermediate result of the correspondence map along with the last colorized frame, the colorization subnet predicts the color for the the current frame.

Colorization Subnet. The correspondence is not accurate everywhere, thus we employ the colorization network $\mathcal{C}$ which is parameterized by $\theta_{\mathcal{C}}$, to select the well-matched colors and propagate them properly. The network receives four inputs: the grayscale input x_t^l, the warped color map $\mathcal{W}^{ab}$ and the confidence map $\mathcal{S}$, and the colorized previous frame $\tilde{x}_{t-1}^{lab}$. Given these, this network predicts the predicted color map $\tilde{x}_t^{ab}$ for the current frame at t:

$$\tilde{x}_t^{ab} = \mathcal{C}(x_t^l, \mathcal{W}^{ab}, \mathcal{S} | \tilde{x}_{t-1}^{lab}; \theta_{\mathcal{C}}) \tag{7}$$

Along with the luminance channel x_t^l, we obtain the colorized image $\tilde{x}_t^{lab}$, also denoted as $\tilde{x}_t$.

3.3. Loss

Our network is supposed to produce realistic video colorization without temporal flickering. Furthermore, the colorization style should resemble the reference in the corresponding regions. To accomplish these objectives, we impose the following losses.

Perceptual Loss. First, to encourage the output to be perceptually plausible, we adopt the *perceptual loss* [44] which measures the semantic difference between the output $\tilde{x}$ and the ground truth image x:

$$\mathcal{L}_{perc} = \|\Phi_{\tilde{x}}^L - \Phi_x^L\|_2^2 \tag{8}$$

where Φ^L represent the feature maps extracted at the $reluL\_2$ layer from the VGG19 network. Here we set $L = 5$ since the top layer captures mostly semantic information. This loss encourages network to select the confident colors from $\mathcal{W}^{ab}$ and propagate them properly.

Contextual Loss. We introduce a *contextual loss*, to encourage colors in the output to be close to those in the reference. The contextual loss is proposed in [45] to measure the local feature similarity while considering the context of the entire image, so it is suitable for transferring the color from the semantically related regions. Our work is the first to apply the contextual loss into exemplar-based colorization. The cosine distances $d^L(i,j)$ are first computed between each pair of feature points $\Phi_{\tilde{x}}^L(i)$ and $\Phi_y^L(j)$, and then normalized as $\tilde{d}^L(i,j) = d^L(i,j)/(\min_k d^L(i,k) + \epsilon), \epsilon = 1e-5$. The pairwise affinities $A^L(i,j)$ between features are defined as:

$$A^L(i,j) = \underset{j}{\text{softmax}}(1 - \tilde{d}^L(i,j)/h) \tag{9}$$

where we set the bandwidth parameter $h = 0.1$ as a recommendation. The affinities $A^l(i,j)$ range within $[0,1]$ and measure the similarity of $\tilde{x}_t(i)$ and $y(j)$ with the Lth layer features. Contrary to the backward matching in [45], we use forward matching where for each feature $\Phi_{\tilde{x},i}^l$ we find the closest feature $\Phi_{\tilde{y},j}^l$ in y. This is because some objects in x_t^l may not exist in y. Consequently, the contextual loss is defined to maximize the affinities between the result and the reference:

$$\mathcal{L}_{context} = \sum_l w_L \left[-\log \left(\frac{1}{N_L} \sum_i \max_j A^L(i,j) \right) \right]. \tag{10}$$

Here we use multiple feature maps: $L = 2$ to 5. N_L denotes the feature number of layer L. We set higher weights w_L for higher level features as the correspondence is proven more reliable using the coarse-to-fine searching strategy [31].

Smoothness Loss. We introduce a *smoothness loss* to encourage spatial smoothness. We assume that neighboring pixels of $\tilde{x}_t$ should be similar if they have similar chrominance in the ground truth image x_t. The smoothness loss is defined as the difference between the color of current pixel and the weighted color of its 8-connected neighborhoods:

$$\mathcal{L}_{smooth} = \frac{1}{N} \sum_{c \in \{a,b\}} \sum_{i} \left(\tilde{x}_t^c(i) - \sum_{j \in \mathbb{N}(i)} w_{i,j} \tilde{x}_t^c(j) \right) \tag{11}$$

where $w_{i,j}$ is the WLS weight [46] which measures the neighborhood correlations. This edge-aware weight helps to produce edge-preserving colorization and alleviate color bleeding artifacts.

Adversarial Loss. We also employ an *adversarial loss* to constrain the colorization video frames to remain realistic. Instead of using image discriminator, a video discriminator is used to evaluate consecutive video frames. We assume that flickering and defective videos can be easily distinguished from real ones, so the colorization network can learn to generate coherent natural results during the adversarial training.

It is tricky to stabilize the adversarial training especially on a large-scale dataset like ImageNet. In this work we adopt the relativistic discriminator [47] which estimates the extent in which the real frames (denoted as z_{t-1} and z_t) look more realistic than the colorized ones $\tilde{x}_{t-1}$ and $\tilde{x}_t$. We adopt the least squares GAN in its relativistic format and the loss for the generator G is defined as:

$$\begin{aligned} \mathcal{L}_{adv}^G = & \mathbb{E}_{(\tilde{x}_{t-1}, \tilde{x}_t) \sim \mathcal{P}_{\tilde{x}}} [(D(\tilde{x}_{t-1}, \tilde{x}_t) \\ & - \mathbb{E}_{(z_{t-1} z_t) \sim \mathcal{P}_z} D(z_{t-1}, z_t) - 1)^2] \\ & + \mathbb{E}_{(z_{t-1} z_t) \sim \mathcal{P}_z} [(D(z_{t-1}, z_t) \\ & - \mathbb{E}_{(\tilde{x}_{t-1}, \tilde{x}_t) \sim \mathcal{P}_{\tilde{x}}} D(\tilde{x}_{t-1}, \tilde{x}_t) + 1)^2] \end{aligned} \tag{12}$$

The relative discriminator loss can be defined in a similar way (see Supplementary Material). From our experiments, this GAN is better to stabilize training than a standard GAN.

Temporal Consistency Loss. To efficiently consider temporal coherency, we also impose a *temporal consistency loss* which explicitly penalizes the color change along the flow trajectory:

$$\mathcal{L}_{temporal} = \| m_{t-1}(p) \odot W_{t-1,t}(\tilde{x}_t^{ab}(p)) - m_{t-1}(p) \odot \tilde{x}_t^{ab}(p) \| \tag{13}$$

where $W_{t-1,t}$ is the forward flow from the last frame x_{t-1} to x_t and m_{t-1} is the binary mask which excludes the occlusion, and $\odot$ represents the Hadamard product.

Figure 3. Augmented training images from ImageNet dataset.

L1 Loss. With the above loss functions, the network can already generate high quality plausible colorized results given a customized reference. Still, we want the network degenerate to the case where the reference comes from the same scene as the video frames. This is a common case for video colorization applications. In this case, we have the ground truth of the predicted frame, so add one more *L1 loss* term to measure the color difference between output $\tilde{x}_t$ and the ground truth x_t:

$$\mathcal{L}_{L1} = \| \tilde{x}_t^{ab} - x_t^{ab} \|_1 \tag{14}$$

Objective Function. Combined with all the above losses, and the overall objective we aim to optimize is:

$$\begin{aligned} \mathcal{L}_I = & \lambda_{perc} \mathcal{L}_{perc} + \lambda_{context} \mathcal{L}_{context} + \lambda_{smooth} \mathcal{L}_{smooth} \\ & + \lambda_{adv} \mathcal{L}_{adv} + \lambda_{L1} \mathcal{L}_{L1} \end{aligned} \tag{15}$$

where λ controls the relative importance of terms. With the guidance of these losses, we successfully unify the correspondence and color propagation within a single network, which learns to generate plausible results based on the exemplar image.

4. Implementation

Network Structure. The correspondence network involves 4 residual blocks each with 2 *conv* layers. The colorization subnet adopts an auto-encoder structure with skip-connections to reuse the low-level features. There are 3 convolutional blocks in the contractive encoder and 3 convolutional blocks in the decoder which recovers the resolution; each convolutional block contains 2$\sim$3 *conv* layers. The $tanh$ serves as the last layer to bound the chrominance output within the color space. The video discriminator consists of 7 *conv* layers where the first six layers halve the input resolution progressively. Also, we insert the self-attention block [48] after the second *conv* layer to let the discriminator examine the global consistency. We use instance normalization since colorization should not be affected by the samples in the same batch. To further improve training stability we apply spectral normalization [49] on both generator and discriminator as suggested in [48].

Training. In order to cover a wide range of scenes, we use multiple datasets for training. First, we collect 1052 videos

from Videvo stock [50] which mainly contains animals and landscapes. Furthermore, we include more portraits videos using the Hollywood2 dataset [51]. We filter out the videos that are either too dark or too faded in color, leaving 768 videos for training. For each video clip we provide reference candidates by inquiring the five most similar images from the corresponding class in the ImageNet dataset. We extract 25 frames from each video and use FlowNet2 [52] to compute the optical flow required for the temporal consistency loss and use the method [53] for the occlusion mask. To further expand the data category, we include images in the ImageNet and apply random geometric distortion and luminance noises to generate augmented video frames as shown in Figure 3. Thus, we get 70k augmented videos in diverse categories. To suit the standard aspect ratio 16:9, we crop all the training images to 384×216. We occasionally provide the reference which is the ground truth image itself but insert Gaussian noise, or feature noise to the VGG features before feeding them into the correspondence network. We deliberately cripple the color matching during training, so the colorization network better learns the color propagation even when the correspondence is inaccurate.

We set $\lambda_{perc} = 0.001$, $\lambda_{context} = 0.2$, $\lambda_{smooth} = 5.0$, $\lambda_{adv} = 0.2$, $\lambda_{flow} = 0.02$ and $\lambda_{L1} = 2.0$. We use a learning rate of 2×10^{-4} for both generator and discriminator without any decay schedule and train the network using the AMSGrad solver with parameters $\beta_1 = 0.5$ and $\beta_2 = 0.999$. We train the network for 10 epochs with a batch size of 40 pairs of video frames.

5. Experiments

In this section, we first study the effectiveness of individual components in our method. Then, we compare our method with state-of-the-art approaches.

5.1. Ablation Studies

Correspondence Learning. To demonstrate the importance of learning parameters in the correspondence subnet, we compare our method with nearest neighbor (NN) matching, in which each feature point of the input image will be matched to the nearest neighbor of the reference feature. Figure 4 shows that our learning-based method matches mostly correct colors from the reference and eases color propagation for the colorization subnet.

Analysis of Loss Functions. We ablate the loss functions individually and evaluate their importance, as shown in Figure 5. When we remove $\mathcal{L}_{perc}$, the colorization fully adopts the color from the reference, but tends to produce more artifacts since there is no loss function to constrain the output semantically similar to the input. When we remove $\mathcal{L}_{context}$, the output does not resemble the reference style. When $\mathcal{L}_{smooth}$ is ablated, colors may not be fully propagated to the whole coherent region. Without $\mathcal{L}_{adv}$, the color appears washed out and perceptually unrealistic. This is because color warping is not accurate and the final output becomes the local color average of the warping color. In comparison, our full model produces vivid colorization with fewer artifacts.

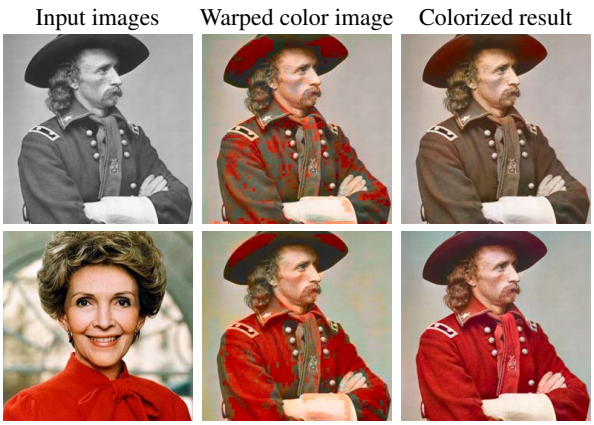

Figure 4. First row: nearest neighbor matching. Second row: with learning parameters in the correspondence network. The first columns list the grayscale image and reference image respectively.

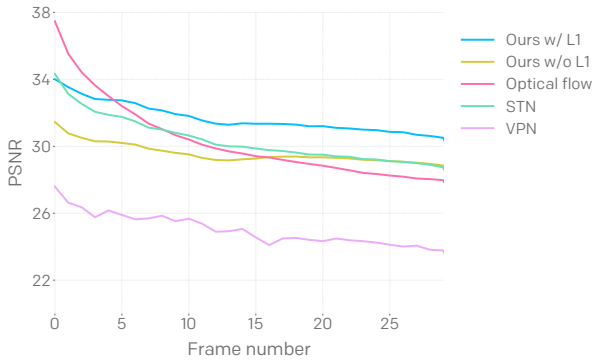

Figure 7. Quantitative comparison on video color propagation.

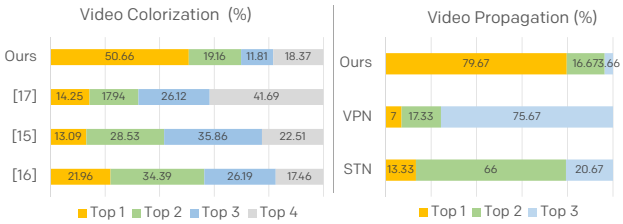

Figure 8. User study results.

5.2. Comparisons

Comparison on Image Colorization. We compare our method against recent learning based image colorization methods both quantitatively and qualitatively. The baseline methods include three automatic colorization methods (Iizuka et al. [15], Larsson et al. [16] and Zhang et al. [17])

Figure 5. Ablation study for different loss functions.

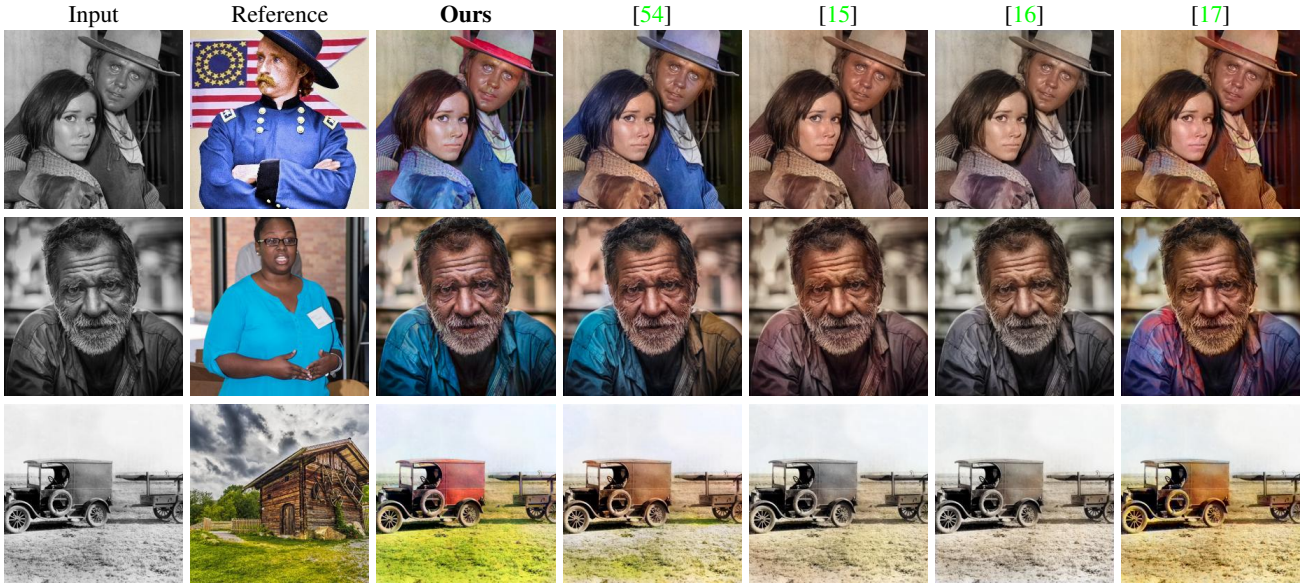

Figure 6. Comparison on image colorization with state-of-the-art methods.

	Top-5 Acc(%)	Top-1 Acc(%)	FID	Colorful	Flicker
GT	90.27	71.19	0.00	19.1	5.22
[15]	85.03	62.94	7.04	11.17	7.19/5.69+
[16]	84.76	62.53	7.26	10.47	6.76/5.42+
[17]	83.88	60.34	8.38	**20.16**	7.93/5.89+
[30]	85.08	64.05	4.78	15.63	NA
Ours	**85.82**	**64.64**	**4.02**	17.90	5.84

Table 1. Comparison with image and per-frame video colorization methods (image test dataset: ImageNet 10k and video test dataset: Videvo.)

and one exemplar based method (He and Chen et al. [30]) since these methods are regarded as state-of-the-art.

For the quantitative comparison, we test these methods on 10k subset of the ImageNet dataset, as shown in Table 1. For exemplar based methods, we take the Top-1 recommendation from ImageNet as the reference. First, we measure the classification accuracy using the VGG19 pre-trained on color images. Our method gives the best *Top-5* and *Top-1* class accuracy, indicating that our method produces semantically meaningful results. Second, we employ the *Fréchet Inception Distance* (FID) [55] to measure the semantic distance between the colorized output and the realistic natural images. Our method achieves the lowest FID, showing that our method provides the most realistic results. In addition, we measure the colorfulness using the psychophysics metric from [56] due to the fact that the users usually prefer colorful images. Table 1 shows that Zhang et al.'s work [17] produces the most vivid color since it encourages rare colors in the loss function; however their method tends to produce visual artifacts, which are also reflected in FID score and the user study. Overall, the results of our method, though slightly less vibrant, exhibit similar colorfulness to the ground truth. The qualitative comparison (in Figure 6) also indicates that our method produces the most realistic, vibrant colorization results.

Comparison with Automatic Video Colorization. In this experiment, we test video colorization on 116 video clips collected from Videvo. We apply the learning based methods for video colorization. It is too costly to use the method in [30] ($> 30s$ whereas $0.61s$ in our method), so we exclude it in this comparison. The quantitative comparison is included in Table 1. We also apply the method proposed in [22] which takes per-frame colorized videos and generate temporally consistent results. We denote these

Figure 9. Comparison of video color propagation. With a given color frame as start, colors are propagated to the succeeding video frames. While other methods purely rely on color propagation, our method takes the initial color frame as a reference and is able to propagate colors for longer interval.

Figure 10. Comparison of automatic video colorization.

post-processed outputs with + in Table 1. We measure the temporal stability using Eq. 13 averaged over all frame pairs in the results. A smaller temporal error represents less flickering. The post-processing method [22] significantly reduces the temporal flickering while our method produces a comparably stable result. However, their method [22] degrades the visual quality since the temporal filtering introduces blurriness. As shown in the example in Figure 10, our method exhibits vibrant colors in each frame with significantly fewer artifacts compared to other methods. Meanwhile, the successively colorized frames demonstrate good temporal consistency.

Comparison with Color Propagation Methods. In order to show that our method can degenerate to the case where the reference is a colored frame for the video itself, we compare it with two recent color propagation methods: VPN [26] and STN [28]. We also include optical flow based color propagation as a baseline. Figure 7 shows the PSNR curve with frame propagation tested on the DAVIS dataset [57]. Optical flow provides the highest PSNR in the initial frames but deteriorates significantly thereafter. The methods STN and VPN also suffer from PNSR degradation. Our method with $\mathcal{L}_1$ loss attains a most stable curve, showing the capability for propagating to longer frames.

User Studies. We first compare our video colorization with three methods of per-frame automatic video colorization: Larsson et al. [16], Zhang et al. [17] and Iizuka et al. [15]. We used 19 videos randomly selected from the Videvo test dataset. For each video, we ask the user to rank the results generated by these four methods in terms of temporal consistency and visual photorealism. Figure 8 (left) shows the results based on the feedback from 20 users. Our

approach is 50.66% more likely to be chosen as the 1st-rank result. Secondly, we compare against two video propagation methods: VPN [26] and STN [28] on 15 randomly selected videos from the DAVIS test dataset. For a fair comparison, we initialize all three methods with the same colorization result of the first frame (using the ground truth video). Figure 8 (right) shows the survey results. Again, our method achieved the highest 1st-rank percentage at 79.67%.

6. Conclusion

In this work, we propose the first exemplar-based video colorization. We unify the semantic correspondence and colorization into a single network, training it end-to-end. Our method produces temporal consistent video colorization with realistic effects. Readers could refer to our supplementary material for more quantitative results.

References

[1] A. Levin, D. Lischinski, and Y. Weiss, "Colorization using optimization," in *ACM transactions on graphics (TOG)*, vol. 23, pp. 689–694, ACM, 2004. 1, 2

[2] L. Yatziv and G. Sapiro, "Fast image and video colorization using chrominance blending," 2004. 1, 2

[3] Y.-C. Huang, Y.-S. Tung, J.-C. Chen, S.-W. Wang, and J.-L. Wu, "An adaptive edge detection based colorization algorithm and its applications," in *Proceedings of the 13th annual ACM international conference on Multimedia*, pp. 351–354, ACM, 2005. 1, 2

[4] Y. Qu, T.-T. Wong, and P.-A. Heng, "Manga colorization," in *ACM Transactions on Graphics (TOG)*, vol. 25, pp. 1214–1220, ACM, 2006. 1, 2

[5] Q. Luan, F. Wen, D. Cohen-Or, L. Liang, Y.-Q. Xu, and H.-Y. Shum, "Natural image colorization," in *Proceedings of the 18th Eurographics conference on Rendering Techniques*, pp. 309–320, Eurographics Association, 2007. 1, 2

[6] T. Welsh, M. Ashikhmin, and K. Mueller, "Transferring color to greyscale images," in *ACM Transactions on Graphics (TOG)*, vol. 21, pp. 277–280, ACM, 2002. 1, 2

[7] A. Bugeau, V.-T. Ta, and N. Papadakis, "Variational exemplar-based image colorization," *IEEE Transactions on Image Processing*, vol. 23, no. 1, pp. 298–307, 2014. 1, 2

[8] X. Liu, L. Wan, Y. Qu, T.-T. Wong, S. Lin, C.-S. Leung, and P.-A. Heng, "Intrinsic colorization," in *ACM Transactions on Graphics (TOG)*, vol. 27, p. 152, ACM, 2008. 1, 2

[9] A. Y.-S. Chia, S. Zhuo, R. K. Gupta, Y.-W. Tai, S.-Y. Cho, P. Tan, and S. Lin, "Semantic colorization with internet images," in *ACM Transactions on Graphics (TOG)*, vol. 30, p. 156, ACM, 2011. 1, 2

[10] R. K. Gupta, A. Y.-S. Chia, D. Rajan, E. S. Ng, and H. Zhiyong, "Image colorization using similar images," in *Proceedings of the 20th ACM international conference on Multimedia*, pp. 369–378, ACM, 2012. 1, 2

[11] G. Charpiat, M. Hofmann, and B. Schölkopf, "Automatic image colorization via multimodal predictions," in *European conference on computer vision*, pp. 126–139, Springer, 2008. 1, 2

[12] R. Ironi, D. Cohen-Or, and D. Lischinski, "Colorization by example.," in *Rendering Techniques*, pp. 201–210, Citeseer, 2005. 1, 2

[13] Y.-W. Tai, J.-Y. Jia, and C.-K. Tang, "Local color transfer via probabilistic segmentation by expectation-maximization," in *IEEE Conference on Computer Vision & Pattern Recognition (CVPR)*, 2005. 1, 2

[14] Z. Cheng, Q. Yang, and B. Sheng, "Deep colorization," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 415–423, 2015. 1, 2

[15] 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 Transactions on Graphics (TOG)*, vol. 35, no. 4, p. 110, 2016. 1, 2, 6, 7, 8

[16] G. Larsson, M. Maire, and G. Shakhnarovich, "Learning representations for automatic colorization," in *European Conference on Computer Vision*, pp. 577–593, Springer, 2016. 1, 2, 6, 7, 8

[17] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," in *European Conference on Computer Vision*, pp. 649–666, Springer, 2016. 1, 2, 6, 7, 8

[18] A. Deshpande, J. Rock, and D. Forsyth, "Learning large-scale automatic image colorization," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 567–575, 2015. 1, 2

[19] J. Zhao, L. Liu, C. G. Snoek, J. Han, and L. Shao, "Pixel-level semantics guided image colorization," *arXiv preprint arXiv:1808.01597*, 2018. 1, 2

[20] F. Baldassarre, D. G. Morín, and L. Rodés-Guirao, "Deep koalarization: Image colorization using cnns and inception-resnet-v2," *arXiv preprint arXiv:1712.03400*, 2017. 1, 2

[21] N. Bonneel, J. Tompkin, K. Sunkavalli, D. Sun, S. Paris, and H. Pfister, "Blind video temporal consistency," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 6, p. 196, 2015. 1, 2

[22] W.-S. Lai, J.-B. Huang, O. Wang, E. Shechtman, E. Yumer, and M.-H. Yang, "Learning blind video temporal consistency," *arXiv preprint arXiv:1808.00449*, 2018. 1, 2, 7, 8

[23] B. Sheng, H. Sun, M. Magnor, and P. Li, "Video colorization using parallel optimization in feature space," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 3, pp. 407–417, 2014. 1, 2

[24] P. Doğan, T. O. Aydın, N. Stefanoski, and A. Smolic, "Key-frame based spatiotemporal scribble propagation," in *Proceedings of the Eurographics Workshop on Intelligent Cinematography and Editing*, pp. 13–20, Eurographics Association, 2015. 1, 2

[25] S. Paul, S. Bhattacharya, and S. Gupta, "Spatiotemporal colorization of video using 3d steerable pyramids," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 8, pp. 1605–1619, 2017. 1, 2

[26] V. Jampani, R. Gadde, and P. V. Gehler, "Video propagation networks," in *Proc. CVPR*, vol. 6, p. 7, 2017. 1, 2, 8, 9

[27] C. Vondrick, A. Shrivastava, A. Fathi, S. Guadarrama, and K. Murphy, "Tracking emerges by colorizing videos," in *Proc. ECCV*, 2018. 1, 2

[28] S. Liu, G. Zhong, S. De Mello, J. Gu, V. Jampani, M.-H. Yang, and J. Kautz, "Switchable temporal propagation network," *arXiv preprint arXiv:1804.08758*, 2018. 1, 2, 8, 9

[29] S. Meyer, V. Cornillère, A. Djelouah, C. Schroers, and M. Gross, "Deep video color propagation," *arXiv preprint arXiv:1808.03232*, 2018. 1, 2

[30] M. He, D. Chen, J. Liao, P. V. Sander, and L. Yuan, "Deep exemplar-based colorization," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 4, p. 47, 2018. 2, 7

[31] J. Liao, Y. Yao, L. Yuan, G. Hua, and S. B. Kang, "Visual attribute transfer through deep image analogy," *arXiv preprint arXiv:1705.01088*, 2017. 2, 4

[32] R. 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," *arXiv preprint arXiv:1705.02999*, 2017. 2

[33] M. He, J. Liao, L. Yuan, and P. V. Sander, "Neural color transfer between images," *arXiv preprint arXiv:1710.00756*, 2017. 2

[34] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *arXiv preprint*, 2017. 2

[35] A. Deshpande, J. Lu, M.-C. Yeh, M. J. Chong, and D. A. Forsyth, "Learning diverse image colorization.," in *CVPR*, pp. 2877–2885, 2017. 2

[36] S. Messaoud, D. Forsyth, and A. G. Schwing, "Structural consistency and controllability for diverse colorization," *arXiv preprint arXiv:1809.02129*, 2018. 2

[37] S. Guadarrama, R. Dahl, D. Bieber, M. Norouzi, J. Shlens, and K. Murphy, "Pixcolor: Pixel recursive colorization," *arXiv preprint arXiv:1705.07208*, 2017. 2

[38] A. Royer, A. Kolesnikov, and C. H. Lampert, "Probabilistic image colorization," *arXiv preprint arXiv:1705.04258*, 2017. 2

[39] V. G. Jacob and S. Gupta, "Colorization of grayscale images and videos using a semiautomatic approach," in *Image Processing (ICIP), 2009 16th IEEE International Conference on*, pp. 1653–1656, IEEE, 2009. 2

[40] N. Ben-Zrihem and L. Zelnik-Manor, "Approximate nearest neighbor fields in video," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5233–5242, 2015. 2

[41] S. Xia, J. Liu, Y. Fang, W. Yang, and Z. Guo, "Robust and automatic video colorization via multiframe reordering refinement," in *Image Processing (ICIP), 2016 IEEE International Conference on*, pp. 4017–4021, IEEE, 2016. 2

[42] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014. 3

[43] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," *arXiv preprint arXiv:1711.07971*, vol. 10, 2017. 3

[44] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *European Conference on Computer Vision*, pp. 694–711, Springer, 2016. 4

[45] R. Mechrez, I. Talmi, and L. Zelnik-Manor, "The contextual loss for image transformation with non-aligned data," *arXiv preprint arXiv:1803.02077*, 2018. 4

[46] Z. Farbman, R. Fattal, D. Lischinski, and R. Szeliski, "Edge-preserving decompositions for multi-scale tone and detail manipulation," in *ACM Transactions on Graphics (TOG)*, vol. 27, p. 67, ACM, 2008. 5

[47] A. Jolicoeur-Martineau, "The relativistic discriminator: a key element missing from standard gan," *arXiv preprint arXiv:1807.00734*, 2018. 5

[48] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," *arXiv preprint arXiv:1805.08318*, 2018. 5

[49] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," *arXiv preprint arXiv:1802.05957*, 2018. 5

[50] "Videvo." https://www.videvo.net/. 6

[51] M. Marszałek, I. Laptev, and C. Schmid, "Actions in context," in *IEEE Conference on Computer Vision & Pattern Recognition*, 2009. 6

[52] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "Flownet 2.0: Evolution of optical flow estimation with deep networks," in *IEEE conference on computer vision and pattern recognition (CVPR)*, vol. 2, p. 6, 2017. 6

[53] M. Ruder, A. Dosovitskiy, and T. Brox, "Artistic style transfer for videos," in *German Conference on Pattern Recognition*, pp. 26–36, Springer, 2016. 6

[54] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016. 7

[55] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," in *Advances in Neural Information Processing Systems*, pp. 6626–6637, 2017. 7

[56] D. Hasler and S. E. Suesstrunk, "Measuring colorfulness in natural images," in *Human vision and electronic imaging VIII*, vol. 5007, pp. 87–96, International Society for Optics and Photonics, 2003. 7

[57] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 724–732, 2016. 8