Semi-Supervised Video Inpainting with Cycle Consistency Constraints

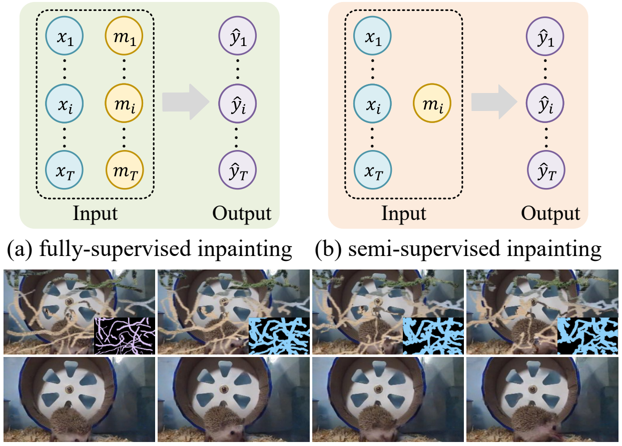
Zhiliang Wu[1], Hanyu Xuan[2]*, Changchang Sun[3], Weili Guan[4,5], Kang Zhang[1], Yan Yan[3]
[1] School of Computer Science and Engineering, Nanjing University of Science and Technology, China
[2] School of Big Data and Statistics, Anhui University, China
[3] Department of Computer Science, Illinois Institute of Technology, USA
[4] School of Information Technology, Monash University, Australia [5] Peng Cheng Laboratory, China

Abstract

Deep learning-based video inpainting has yielded promising results and gained increasing attention from researchers. Generally, these methods assume that the corrupted region masks of each frame are known and easily obtained. However, the annotation of these masks are labor-intensive and expensive, which limits the practical application of current methods. Therefore, we expect to relax this assumption by defining a new semi-supervised inpainting setting, making the networks have the ability of completing the corrupted regions of the whole video using the annotated mask of only one frame. Specifically, in this work, we propose an end-to-end trainable framework consisting of completion network and mask prediction network, which are designed to generate corrupted contents of the current frame using the known mask and decide the regions to be filled of the next frame, respectively. Besides, we introduce a cycle consistency loss to regularize the training parameters of these two networks. In this way, the completion network and the mask prediction network can constrain each other, and hence the overall performance of the trained model can be maximized. Furthermore, due to the natural existence of prior knowledge (e.g., corrupted contents and clear borders), current video inpainting datasets are not suitable in the context of semi-supervised video inpainting. Thus, we create a new dataset by simulating the corrupted video of real-world scenarios. Extensive experimental results are reported to demonstrate the superiority of our model in the video inpainting task. Remarkably, although our model is trained in a semi-supervised manner, it can achieve comparable performance as fully-supervised methods.

1. Introduction

Video inpainting aims to fill corrupted regions of the video with plausible contents, which is a promising yet challenging task in computer vision. It can benefit a wide range of practical applications, such as scratch restoration [1], undesired object removal [30], and autonomous driving [18], etc. In essence, unlike image inpainting which usually learns on the spatial dimension, video inpainting task pays more attention to exploiting the temporal information. Naively using image inpainting methods [13, 34, 42, 45] on individual video frame to fill corrupted regions will lose inter-frame motion continuity, resulting in flicker artifacts in the inpainted video.



(a) fully-supervised inpainting (b) semi-supervised inpainting

(c) an example of semi-supervised video inpainting

Figure 1. Existing methods perform video inpainting in a fully-supervised setting. The typical issue of such methods is the need to elaborately annotate the corrupted regions $\boldsymbol{m}_i$ of each frame $\boldsymbol{x}_i$ in the video (Fig.(a)), which is labor-intensive and expensive in real-world applications. In this paper, we formulate a new task: semi-supervised video inpainting, which only annotates the corrupted regions of one frame to complete the whole video (Fig.(b)). Fig.(c) shows an example of semi-supervised video inpainting: the top row shows sample frames with the mask, where pink denotes the manually annotated mask of corrupted regions, and blue denotes the mask of corrupted regions predicted by the model. The completed results $\widehat{\boldsymbol{y}}_i$ are shown in the bottom row.

Similar to the texture synthesis, traditional video inpaint-

*Corresponding author

ing methods [5, 8, 23] attempt to generate missing contents to fill missing regions by searching and copying similar patches from known regions. Despite the authentic completion results have been achieved, these approaches still meet grand challenges, such as the lack of high-level understanding of the video [32] and high computational complexity [1, 16]. In fact, recently, deep learning-based video inpainting methods [1,4,11,16,17,26,32,41,50] have been noticed by many researchers and made promising progress in terms of the quality and speed. These methods usually extract relevant features from the neighboring frames by convolutional neural networks and perform different types of context aggregation to generate missing contents. Nevertheless, these methods are only suitable in the scenarios where the corrupted region mask of each frame in the video is available (Fig.1(a)). Hence, when we resort to these methods in real-world applications, annotating the corrupted regions of each frame in the video is required, which is labor-intensive and expensive, especially for long videos. Formally, a naive combination of video object segmentation and video inpainting methods can be used to reduce annotation costs in a two-stage manner. In this way, we can first generate the masks for all video frames using a video object segmentation method, and then complete the missing regions with the fully-supervised video inpainting methods. However, such a two-stage approach has some intuitive disadvantages. One the one hand, since each module is learned individually and sometimes not well combined to maximize performance, the overall performance is sub-optimal. On the other hand, existing video object segmentation methods are unsatisfactory for segmentation results of the scratch regions similar to Fig.1(c), resulting in the inpainted video with critical errors. Therefore, to realize the goal of reducing annotation cost for video inpainting from scratch, in this paper, we introduce a new semi-supervised video inpainting setting that we can complete the corrupted regions of the whole video using the annotated mask of only one frame (Fig.1(b)) and train the network from end-to-end. In this way, compared with the conventional fully-supervised setting, the annotation cost can be greatly reduced, making video inpainting more convenient in practical application.

However, fulfilling the task of semi-supervised video inpainting is non-trivial and has some issues to be addressed. On the one hand, except for one annotated known frame, there are no masks for other frames to indicate the corrupted regions in the proposed semi-supervised setting. To solve this problem, we decompose the semi-supervised video inpainting task into dual tasks: frame completion and mask prediction. Specifically, we first perform frame completion on the frame with corresponding given mask to obtain the completed frame using the designed frame completion network. Then, we feed the completed frame and the subsequent frame into the proposed mask prediction network to generate the corrupted region mask of the subsequent frame. Last, by iterating frame by frame, we can complete corrupted regions of each frame in the video. On the other hand, to precisely capture the accurate correspondence between the completion network and mask prediction network, a cycle consistency loss is introduced to regularize the trained parameters. In addition, existing video inpainting datasets usually take black or noise pixels as the corrupted contents of the video frame. In fact, such a setting will introduce some specific prior knowledge (e.g., corrupted contents and clear borders) into the dataset, making it easy for the mask prediction network to distinguish corrupted regions from natural images. In this way, existing datasets cannot realistically simulate complex real-world scenarios. Hence, in our work, to effectively avoid the introduction of the above prior knowledge into the dataset, we use natural images as corrupted contents of the video frame and apply iterative Gaussian smoothing [35] to extend the edges of corrupted regions. Experimental results demonstrate that our proposed method can achieve comparable inpainting results as fully-supervised methods. An example result of our method is shown in Fig.1(c).

Our contributions are summarized as follows:

- We formulate a novel semi-supervised video inpainting task that aims to complete the corrupted regions of the whole video with the given mask of one frame. To the best of our knowledge, this is the first end-to-end semi-supervised work in the video inpainting field.
- A flexible and efficient framework consisting of completion network and mask prediction network is designed to solve the semi-supervised video inpainting task, where cycle consistency loss is introduced to regularize the trained parameters.
- A novel synthetic dataset[1] is tailored for the semi-supervised video inpainting task. which consists of 4,453 video clips. This dataset will be published to facilitate subsequent research and benefit other researchers.

2. Related Works

Video Inpainting can be roughly classified into two lines: patch-based and deep learning-based video inpainting.

Patch-based inpainting methods [5, 8, 23] solve the video inpainting task by searching and pasting coherent contents from the known regions into the corrupted regions. However, these approaches often suffer high computational complexity due to the heavy optimization process, which limits their real-world applications [1].

[1] https://drive.google.com/drive/folders/1g_Vb-14DFKrc9-Ao-iTHGtKY4titHFj3.

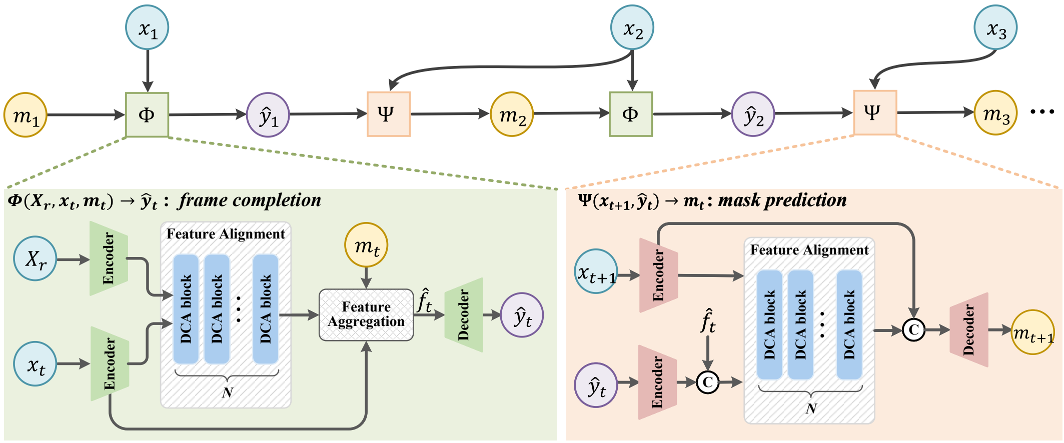


Figure 2. **Illustration of the proposed semi-supervised video inpainting networks.** Our networks are composed of a *completion network*($\boldsymbol{\Phi}$) and a *mask prediction network*($\boldsymbol{\Psi}$). The *completion network* uses temporal information of reference frames $\boldsymbol{X}_r$ to complete the corrupted regions $\boldsymbol{m}_t$ of the target frame $\boldsymbol{x}_t$. The *mask prediction network* takes the completed target frame $\widehat{\boldsymbol{y}}_t$ and frame $\boldsymbol{x}_{t+1}$ as input to generate the mask $\boldsymbol{m}_{t+1}$, which can be adopted to indicate the corrupted regions of the next frame $\boldsymbol{x}_{t+1}$. By alternating forward, we can complete the whole video with only one annotated frame. DCA block denotes deformable convolution alignment block. C stands for the concatenate operation.

Deep learning-based video inpainting mainly focuses on three directions: 1) *3D convolutional networks* [1, 11, 20], which usually reconstruct the corrupted contents by directly aggregating temporal information from neighbor frames through 3D temporal convolution; 2) *flow guided approaches* [4, 10, 15, 17, 41, 47, 50], which first use a deep flow completion network to restore the flow sequence, and then use the restored flow sequence to guide the relevant pixels of the neighbor frames into corrupted regions; and 3) *attention-based methods* [16, 19, 21, 26, 30, 37, 38, 46], which retrieve information from neighbor frames and use weighted sum to generate corrupted contents. Although these methods have achieved promising completion results, we still need to elaborately annotate the corrupted regions of each frame in the video, which limits its applications. Unlike these approaches, our proposed semi-supervised video inpainting only uses the mask of one frame to complete the corrupted regions of the whole video.

Semi-supervised Video Object Segmentation is understood as using a single annotated frame (usually the first frame of the sequence) to estimate the object position in the subsequent frames of the video. Existing semi-supervised video object segmentation can be broadly classified into three categories, *i.e.*, *matching-based* [7, 27, 39], *propagation-based* [2, 9, 48], and *detection*-based methods [31,33,43]. Matching-based methods [7,27,39] usually train a typical Siamese matching network to find objects similar to the given mask of the first frame in subsequent frames. Besides, propagation-based methods [2,9,48] work on embedding image pixels into a feature space and utilized temporal information to guide label propagation. In addition, Detection-based methods [31,33,43] first learn feature representation of the annotated objects in the first frame and then detect corresponding pixels in subsequent frames. In spite of the compelling success achieved by these methods, far too little attention has been paid to the segmentation of scratched regions with complex patterns. In this situation, we put forward a mask prediction network that utilizes the current completed frame and subsequent frame as input to the generate corrupted region mask. Fortunately, we find the fact that the only difference between the video frames before and after completion is the corrupted regions of the current frame. Therefore, our mask prediction network is suitable for predicting damaged regions of various modalities, so that making our proposed semi-supervised inpainting framework robust.

3. Method

3.1. Problem Formulation

Let $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_T\}$ be a corrupted video sequence consisting of T frames. $\boldsymbol{M} = \{\boldsymbol{m}_1, \boldsymbol{m}_2, \ldots, \boldsymbol{m}_T\}$ denotes the corresponding frame-wise masks, where the mask $\boldsymbol{m}_i$ represents the corrupted regions of frame $\boldsymbol{x}_i$. The goal of video inpainting is to predict a completed video $\widehat{\boldsymbol{Y}} = \{\widehat{\boldsymbol{y}}_1, \widehat{\boldsymbol{y}}_2, \ldots, \widehat{\boldsymbol{y}}_T\}$, which should be spatially and temporally consistent with the ground truth video $\boldsymbol{Y} = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_T\}$. Specifically, a mapping function from the input $\boldsymbol{X}$ to the output $\widehat{\boldsymbol{Y}}$ needs to be learned so that the conditional distribution $p(\widehat{\boldsymbol{Y}}|\boldsymbol{X})$ is approximate to $p(\boldsymbol{Y}|\boldsymbol{X})$.

Based on the fact that the contents of the corrupted regions in one frame may exist in neighboring frames, the video inpainting task can be formulated as a conditional pixel prediction problem:

$$p(\widehat{\boldsymbol{Y}}|\boldsymbol{X}) = \prod_{t=1}^{T} p(\widehat{\boldsymbol{y}}_t|\boldsymbol{X}_r, \boldsymbol{x}_t, \boldsymbol{m}_t), \tag{1}$$

where $\boldsymbol{X}_r = \{\boldsymbol{x}_{t-n}, \ldots, \boldsymbol{x}_{t-1}, \boldsymbol{x}_{t+1}, \ldots, \boldsymbol{x}_{t+n}\}$, namely *reference frames*; $\boldsymbol{x}_t$ presents the current frame that needs to be inpainted, namely *target frame*. Formally, the existing video inpainting methods aim to model the conditional distribution $p(\widehat{\boldsymbol{y}}_t|\boldsymbol{X}_r, \boldsymbol{x}_t, \boldsymbol{m}_t)$ by training a deep neural network D, *i.e.*, $\widehat{\boldsymbol{y}}_t = D(\boldsymbol{X}_r, \boldsymbol{x}_t, \boldsymbol{m}_t)$. The final output $\widehat{\boldsymbol{Y}}$ is obtained by processing the video frame by frame in temporal order. Unfortunately, for our semi-supervised video inpainting setting, only the mask $\boldsymbol{m}_i$ of the one frame $\boldsymbol{x}_i$ is provided[2], while the masks of corrupted regions in other frames are unknown. This means that we don't know where other frames need to be inpainted except for the frames of the given mask.

To tackle this issue, we decompose the semi-supervised video inpainting task into a pair of dual tasks: *frame completion* and *mask prediction*. The former aims to generate what to inpaint, while the latter is designed to estimate where to inpaint (via masks). By this means, we can generate complete the current frame, and generate the corrupted regions mask of the next frame as well. Thereafter, by iterating frame by frame, we can complete the corrupted regions of the whole video sequence based on the only one known mask. Conditioned on a corrupted video sequence $\boldsymbol{X}$, the dual tasks can be defined as:

- **Frame completion** aims to learn a mapping $\boldsymbol{\Phi}$ to generate the completed frame $\widehat{\boldsymbol{y}}_t$ of the target frame $\boldsymbol{x}_t$ utilizing the reference frames $\boldsymbol{X}_r$, *i.e.*, $\widehat{\boldsymbol{y}}_t = \boldsymbol{\Phi}(\boldsymbol{X}_r, \boldsymbol{x}_t, \boldsymbol{m}_t)$;

- **Mask prediction** expect to learn a mapping $\boldsymbol{\Psi}$ to inversely generate mask $\boldsymbol{m}_{t+1}$ by using the frame $\boldsymbol{x}_{t+1}$ and the completed frame $\widehat{\boldsymbol{y}}_t$, *i.e.*, $\boldsymbol{m}_{t+1} = \boldsymbol{\Psi}(\boldsymbol{x}_{t+1}, \widehat{\boldsymbol{y}}_t)$.

3.2. Network Design

We design an end-to-end trainable framework to tackle semi-supervised video inpainting task. As shown in Fig.2, our framework consists of a **completion network** and a **mask prediction network**. The former aims to use the temporal information of the reference frames $\boldsymbol{X}_r$ to complete the corrupted regions $\boldsymbol{m}_t$ of the target frame $\boldsymbol{x}_t$, while the latter utilizes the completed target frame $\widehat{\boldsymbol{y}}_t$ to predict the corrupted regions $\boldsymbol{m}_{t+1}$ of the subsequent frame $\boldsymbol{x}_{t+1}$.

3.2.1 Completion Network.

As shown in Fig.2, completion network consists of four parts: frame-level encoder, feature alignment module, feature aggregation module, and frame-level decoder. The frame-level encoder aims to extract deep features from low-level pixels of each frame, while the frame-level decoder is used to decode completed deep features into the frame. They consist of multiple convolutional layers and residual blocks with ReLUs as the activation functions. ***Feature alignment*** and ***feature aggregation*** modules are the core components of the completion network. The former performs reference frame alignment at the feature level to eliminate image changes between the reference frame and the target frame, while the latter aggregates the aligned reference frame features to complete corrupted regions of the target frame.

Feature Alignment. Due to the image variation caused by camera and object motion, it is difficult to directly utilize the temporal information of the reference frames $\boldsymbol{X}_r$ to complete the corrupted regions $\boldsymbol{m}_t$ of the target frame $\boldsymbol{x}_t$. Therefore, an extra alignment module is necessary for video inpainting. Notably, deformable convolution [3] can obtain information away from its regular local neighborhood by learning offsets of the sampling convolution kernels. Motivated by the capacity of deformable convolution, a Deformable Convolution Alignment (DCA) block is designed to perform reference frame alignment at the feature level. Specifically, for the target frame feature $\boldsymbol{f}_t$ and the reference frame feature $\boldsymbol{f}_r$ extracted by the frame-level encoder, we first cascade them to predict the offsets $\theta = \{\triangle \boldsymbol{p}_n | n = 1, \ldots, |R|\}$ of the convolution kernel, where $r \in \{t-n, \ldots, t-1, t+1, \ldots, t+n\}$, and $R = \{(-1,-1), (-1,0), \ldots, (1,1)\}$ denotes a regular grid of a 3×3 kernel. Then, with the predicted offset θ, the aligned feature $\boldsymbol{f}_r^a$ of the reference frame feature $\boldsymbol{f}_r$ is obtained by a deformable convolution layer $\mathcal{DCN}$:

$$\boldsymbol{f}_r^a = \mathcal{DCN}(\boldsymbol{f}_r, \theta). \tag{2}$$

In practice, to facilitate feature alignment of the reference frames more accurately, we cascade four DCA blocks in the feature alignment module to enhance its transformation flexibility and capability. Ablation study on the performance of the feature alignment with different numbers of the DCA blocks can be found in Section. 4.4.

Feature Aggregation. Due to occlusion, blur and parallax problems, different aligned reference frame features are not equally beneficial for the reconstruction of corrupted contents of the target frame. Therefore, an adaptive temporal features aggregation module is introduced to dynamically aggregate aligned reference frame features as shown in Fig.3. Specifically, for each aligned reference frame feature $\boldsymbol{f}_r^a$, we first calculate the aggregate weights $\boldsymbol{s}_r$ by the

[2] Here, we usually assume that the annotated frame is the first frame of the video.

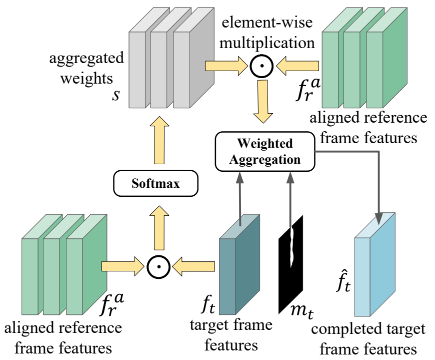


Figure 3. Illustration of adaptive temporal features aggregation module.

softmax function:

$$\boldsymbol{s}_r = \frac{exp\left(\mathcal{Q}(\boldsymbol{f}_t)^T \cdot \mathcal{K}(\boldsymbol{f}_r^a)\right)}{\sum_r exp\left(\mathcal{Q}(\boldsymbol{f}_t)^T \cdot \mathcal{K}(\boldsymbol{f}_r^a)\right)}, \tag{3}$$

where $\mathcal{Q}(\cdot)$ and $\mathcal{K}(\cdot)$ denote 1×1 2D convolution. After obtaining the aggregated weight $\boldsymbol{s}$ for all reference frame features, the modulation feature $\boldsymbol{h}_r$ corresponding to the feature $\boldsymbol{f}_r^a$ is obtained by a pixel-wise manner:

$$\boldsymbol{h}_r = \mathcal{V}(\boldsymbol{f}_r^a) \odot \boldsymbol{s}_r, \tag{4}$$

where $\mathcal{V}(\cdot)$ denotes 1×1 2D convolution, $\odot$ denotes the element-wise multiplication. Finally, the aggregated features $\widehat{\boldsymbol{f}}_t$ are obtained by a fusion convolutional layer:

$$\widehat{\boldsymbol{f}}_t = \mathcal{A}([\boldsymbol{h}_{t-n}, \ldots, \boldsymbol{h}_{t+n}, \boldsymbol{f}_t, \boldsymbol{m}_t]), \tag{5}$$

where $[\cdot,\cdot,\cdot]$ denotes the concatenation operation, $\mathcal{A}$ is a 1×1 convolution layer. The final inpainted target frame $\widehat{\boldsymbol{y}}_t$ corresponding to the target frame $\boldsymbol{x}_t$ can be obtained by decoding $\widehat{\boldsymbol{f}}_t$ with the frame-level decoder.

3.2.2 Mask Prediction Network.

Mask prediction network aims to predict the corrupted regions of the video frame. A naive idea is to use the corrupted regions as a segmentation object to generate corrupted regions mask for subsequent frames by state-of-the-art video object segmentation (VOS) methods. Although VOS methods have achieved significant results in the past few years, their performance is unsatisfactory for segmenting corrupted regions of videos due to the following reasons: 1) the appearance of the corrupted regions varies greatly in the video; 2) the boundary between the corrupted regions and the uncorrupted regions is blurred. Fortunately, we find the fact that the only difference between the video frames before and after completion is the corrupted regions of the current frame.Therefore, we design a mask prediction network that utilizes current completed frame $\widehat{\boldsymbol{y}}_t$ and subsequent frame $\boldsymbol{x}_{t+1}$ as input to generate corrupted region mask $\boldsymbol{m}_{t+1}$ for subsequent frame.

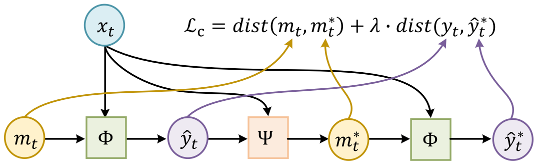

x_t
$\mathcal{L}_c = dist(m_t, m_t^*) + \lambda \cdot dist(y_t, \hat{y}_t^*)$
m_t
Φ
$\hat{y}_t$
Ψ
m_t^*
Φ
$\hat{y}_t^*$

Figure 4. Illustration of cycle consistency.

As shown in Fig.2, mask prediction network consists of three parts: frame-level encoder, feature alignment module , and frame-level decoder. Specifically, we first extract the deep features $\boldsymbol{g}_t$ and $\boldsymbol{f}_{t+1}$ of the current completed frame $\widehat{\boldsymbol{y}}_t$ and subsequent frame $\boldsymbol{x}_{t+1}$ by the frame-level encoder, respectively. Here, to enrich the deep feature $\boldsymbol{g}_t$ of the completed frame $\widehat{\boldsymbol{y}}_t$, we concatenate the completed feature $\widehat{\boldsymbol{f}}_t$ obtained in the completion network with $\boldsymbol{g}_t$, i.e., $\widehat{\boldsymbol{q}}_t = [\boldsymbol{g}_t, \widehat{\boldsymbol{f}}_t]$. Then, the feature alignment module is used to align the concatenated feature $\widehat{\boldsymbol{q}}_t$, which aims to eliminate the effects of image changes between current completed frame $\widehat{\boldsymbol{y}}_t$ and subsequent frame $\boldsymbol{x}_{t+1}$. Note that the feature alignment module of mask prediction network has the same structure as the feature alignment module of completion network, and the parameters between them are shared. Finally, the aligned feature $\widehat{\boldsymbol{q}}_t^a$ and subsequent frame feature $\boldsymbol{f}_{t+1}$ are concatenated into a frame-level decoder to generate the corrupted regions mask $\boldsymbol{m}_{t+1}$ for subsequent frame $\boldsymbol{x}_{t+1}$.

3.3. Cycle Consistency

In our task, the correspondence between $\widehat{\boldsymbol{y}}_t$ and $\boldsymbol{m}_t$ is one-to-one, so the frame completion and mask prediction exist simultaneously. In other words, for video sequence $\boldsymbol{X}$, given the corresponding mask $\boldsymbol{m}_t$ of target frame $\boldsymbol{x}_t$, the completed frame $\widehat{\boldsymbol{y}}_t$ can be obtained by $\widehat{\boldsymbol{y}}_t = \Phi(\boldsymbol{X}_r, \boldsymbol{x}_t, \boldsymbol{m}_t)$. Conversely, given the corresponding completed frame $\widehat{\boldsymbol{y}}_t$ of target frame $\boldsymbol{x}_t$, the mask $\boldsymbol{m}_t$ can be obtained by $\boldsymbol{m}_t = \Psi(\boldsymbol{x}_t, \widehat{\boldsymbol{y}}_t)$. Ideally, if the mapping Φ and Ψ both can capture accurate correspondence, nesting them together can obtain the following cycle consistency:

$$\widehat{\boldsymbol{y}}_t^* = \Phi(\boldsymbol{X}_r, \boldsymbol{x}_t, \Psi(\boldsymbol{x}_t, \widehat{\boldsymbol{y}}_t)), \tag{6}$$

$$\boldsymbol{m}_t^* = \Psi(\boldsymbol{x}_t, \Phi(\boldsymbol{X}_r, \boldsymbol{x}_t, \boldsymbol{m}_t)). \tag{7}$$

Eq.(6) and Eq.(7) give us the solution to generate reliable and consistent correspondence between the mapping Φ and Ψ by formulating the loss as follows,

$$\mathcal{L}_y = dist(\widehat{\boldsymbol{y}}_t, \widehat{\boldsymbol{y}}_t^*), \tag{8}$$

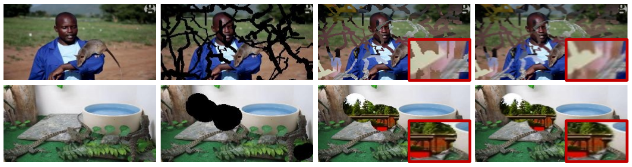

(a) ground truth (b) existing data (c) w/o gaussian (d) ours data

Figure 5. Example of generated training data.

$$\mathcal{L}_m = dist(\boldsymbol{m}_t, \boldsymbol{m}_t^*), \tag{9}$$

where $dist(\cdot,\cdot)$ is L_1 distance function.

As shown in Fig.4, by combining Eq.(8) and Eq.(9), we obtain the hybrid loss as follows,

$$\mathcal{L}_c = \mathcal{L}_m + \lambda_y \mathcal{L}_y, \tag{10}$$

where λ_y is the non-negative trade-off parameter.

3.4. Loss Function

We train our network by minimizing the following loss:

$$\mathcal{L} = \lambda_f \mathcal{L}_f + \lambda_s \mathcal{L}_s + \lambda_c \mathcal{L}_c, \tag{11}$$

where $\mathcal{L}_f$ is the frame reconstruction loss, $\mathcal{L}_s$ is the mask prediction loss and $\mathcal{L}_c$ is the cycle consistency loss. λ_f, λ_s and λ_c are the trade-off parameters. In real implementation, we empirically set the weights of different losses as: $\lambda_f = 2.5$, $\lambda_s = 0.25$ and $\lambda_c = 1$.

4. Experiments

4.1. Dataset and Implementation Details

Training Data Generation. In general, the diversity of the training set is the essential prerequisite for the robustness of the trained model. Existing benchmark datasets are usually generated by $\boldsymbol{x}_i = (1 - \boldsymbol{m}_i) \odot \boldsymbol{y}_i + \boldsymbol{m}_i \odot \boldsymbol{u}_i$, where $\boldsymbol{u}_i$ is a noise signal having the same size as $\boldsymbol{y}_i$, $\odot$ is the element-wise multiplication. Traditional fully-supervised video inpainting tasks usually assume $\boldsymbol{u}_i$ as a constant value or certain kind of noise (Fig.5(b)). However, when $\boldsymbol{u}_i$ is set as a constant value or a certain kind of noise, both $\boldsymbol{u}_i$ and $\boldsymbol{m}_i$ would be easily distinguished by a deep neural net or even a simple linear classifier from a natural image. In this way, video inpainting task is fallen into the fully-supervised category with almost perfect prediction of $\boldsymbol{m}_i$, which can be solved using existing methods. In fact, such an assumption generally does not hold in real-world scenarios.

Therefore, for semi-supervised video inpainting tasks, the key to dataset generation is how to define noise $\boldsymbol{u}_i$ to mask it as different as possible from $\boldsymbol{x}_i$ in image pattern. In this paper, we use real-world image patches to define noisy $\boldsymbol{u}_i$. This definition can ensure that the local patches between the noise $\boldsymbol{u}_i$ and the image $\boldsymbol{x}_i$ cannot be easily distinguished, enforcing the network to infer the location $\boldsymbol{m}_i$ of the noise $\boldsymbol{u}_i$ according to the context of the $\boldsymbol{x}_i$, which eventually improves the generalization ability for real-world data. Furthermore, it is worth noting that the existing methods of fully supervised video inpainting usually use the fixed-shape $\boldsymbol{m}_i$ (*e.g,* rectangle) to generate datasets, which is disadvantageous for our semi-supervised video inpainting task. This is because the fixed-shape $\boldsymbol{m}_i$ introduces prior knowledge into the training of the model, which encourages the mask prediction network to locate corrupted regions based on the mask shape. Therefore, free-form strokes [1] with shape diversity are used as our $\boldsymbol{m}_i$ to generate the dataset, making the mask prediction network harder to infer corrupted regions by using shape information.

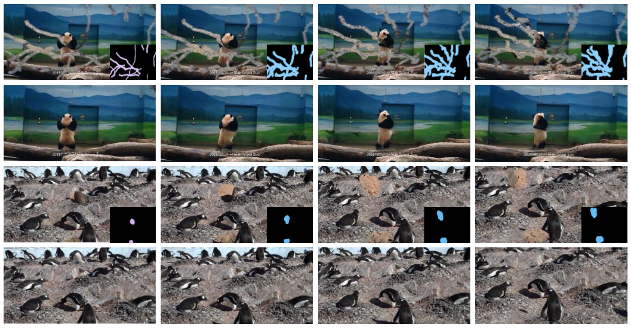


Figure 6. Two example of inpainting results with our method. The top row shows sample frames with the mask, where pink denotes the manually annotated object mask, and blue denotes the segmentation mask generated by the model. The completed results are shown in the bottom row.

In addition, the mixed image $\boldsymbol{x}_i$ obtained directly through $\boldsymbol{x}_i = (1 - \boldsymbol{m}_i) \odot \boldsymbol{y}_i + \boldsymbol{m}_i \odot \boldsymbol{u}_i$ using $\boldsymbol{y}_i$ and $\boldsymbol{u}_i$ would lead to noticeable edges (Fig.5(c)), which are strong indicators for distinguishing corrupted regions. Such edge priors will inevitably sacrifice the semantic understanding capability of the mask prediction network. Therefore, we use iterative Gaussian smoothing [35] to extend $\boldsymbol{m}_i$ in the process of dataset generation, and employ alpha blending in the contact regions between $\boldsymbol{u}_i$ and $\boldsymbol{y}_i$, which effectively avoids introducing obvious edge priors into the dataset. Our dataset is generated on the basis of Youtube-vos [40], *i.e.* video frames in Youtube-vos as our $\boldsymbol{y}_i$. Following the settings of the original Youtube-vos dataset, our generated dataset contains 3471, 474, and 508 video clips in training, validation, and test sets, respectively.

Testing Dataset To evaluate the effectiveness of our method, two popular video object segmentation datasets are taken to evaluate our model, including Youtube-vos [40] and DAVIS [25]. Following the previous settings [37, 41], DAVIS contains 60 video clips. To ensure the comparability of experimental results, all baseline methods used for comparison are fine-tuned multiple times on our generated

Table 1. Quantitative results of video inpainting. The term *Semi* denotes 'Semi-supervised' for short, *Our_ComNet* represents the completion network trained in the fully supervised manner.

		Youtube-vos				DAVIS			
	Semi	PSNR↑	SSIM↑	E_{warp} ↓	LPIPS↓	PSNR↑	SSIM↑	E_{warp} ↓	LPIPS↓
TCCDS	✗	23.418	0.8119	0.3388	1.9372	28.146	0.8826	0.2409	1.0079
VINet	✗	26.174	0.8502	0.1694	1.0706	29.149	0.8965	0.1846	0.7262
DFVI	✗	28.672	0.8706	0.1479	0.6285	30.448	0.8961	0.1640	0.6857
CPVINet	✗	28.534	0.8798	0.1613	0.8126	30.234	0.8997	0.1892	0.6560
FGVC	✗	24.244	0.8114	0.2484	1.5884	28.936	0.8852	0.2122	0.9598
OPN	✗	30.959	0.9142	0.1447	0.4145	32.281	0.9302	0.1661	0.3876
STTN	✗	28.993	0.8761	0.1523	0.6965	28.891	0.8719	0.1844	0.8683
FuseFormer	✗	29.765	0.8876	0.1463	0.5481	29.627	0.8852	0.1767	0.6706
E2FGVI	✗	30.064	0.9004	0.1490	0.5321	31.941	0.9188	0.4579	0.6344
Our_ComNet	✗	**31.291**	**0.9237**	**0.1423**	**0.3918**	**32.807**	**0.9401**	**0.1503**	**0.3681**
RANet+OPN	✓	21.826	0.8058	0.2446	0.9346	20.609	0.8094	0.3251	0.7315
STM+FuseFormer	✓	23.371	0.8283	0.1919	0.8613	19.917	0.8906	0.2569	0.8119
GMN+STTN	✓	22.680	0.8145	0.1822	0.7335	20.394	0.8204	0.2647	0.8071
HMMN+E2FGVI	✓	25.255	0.8539	0.1803	0.7019	22.696	0.8635	0.2429	0.7313
Ours	✓	**30.834**	**0.9135**	**0.1452**	**0.4171**	**32.097**	**0.9263**	**0.1534**	**0.3852**

dataset by their released models and codes, and the best results are reported.

Implementation Details. We use PyTorch to implement our model. The proposed semi-supervised video inpainting network is trained by using Adam optimizer with learning rate 1e-4 and $\beta = (0.9, 0.999)$. The video sequences are resized to 256×256 during the training.

4.2. Video Inpainting Evaluation

Baselines and evaluation metrics. To evaluate the video inpainting ability of our model, nine state-of-the-art video inpainting methods are used as our baselines, including one optimization-based method: TCCDS [8] and eight learning-based methods: VINet [11, 12], DFVI [41], CPVINet [16], FGVC [4], OPN [30], STTN [44], FuseFormer [19], and E2FGVI [17]. Note that there was no work focusing on semi-supervised video inpainting task before, baselines mentioned above are all worked in a fully-supervised manner. Furthermore, to strengthen the baselines in the semi-supervised setting, we also introduce the methods of connecting VOS and VI as our baselines, *i.e.*, using the existing VOS methods (RANet [36], GMN [22], STM [29], and HMMN [28]) to obtain full mask information of the video, then completing missing regions of the video with the fully-supervised VI methods (OPN [30], STTN [44], FuseFormer [19], and E2FGVI [17]). The quantitative results of video inpainting are reported by four metrics, *i.e.*, PSNR [6], SSIM [30], LPIPS [49], and flow warping error E_{warp} [14].

Experimental results and analysis. The quantitative results of video inpainting on Youtube-vos and DAVIS datasets are summarized in Tab.1. It can be seen from Tab.1 that the PSNR, SSIM, E_{warp} and LPIPS of our semi-supervised model are comparable to the fully-supervised baselines on two datasets, and significantly outperform the method concatenating VOS and VI. Fig.6 shows some visual results of semi-supervised video inpainting obtained by our method. It can be seen from Fig.6 that the proposed method can generate the spatio-temporally consistent content in the missing regions of the video. It should be noted that our semi-supervised model only uses the mask of the first frame to indicate corrupted regions when completing the whole video, which is greatly beneficial for the practical application of video inpainting compared to the fully-supervised baselines that requires annotations of all frames.

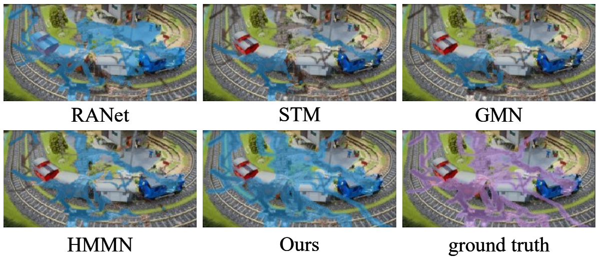

RANet STM GMN
HMMN Ours ground truth

Figure 7. Example of corrupted regions segmentation.

Table 2. Evaluation results of mask prediction.

	Youtube-vos		DAVIS	
	BCE↓	IOU↑	BCE↓	IOU↑
RANet	4.053	0.411	4.112	0.381
GMN	2.381	0.587	4.602	0.363
STM	2.032	0.605	4.851	0.316
HMMN	1.567	0.719	3.738	0.456
Ours	**1.148**	**0.934**	**1.327**	**0.832**

4.3. Mask Prediction Evaluation

Baselines and evaluation metrics. In this section, we use the learned mask prediction network to evaluate the corrupted regions segmentation ability of our model. To intuitively reflect the segmentation ability of our model, the specialized semi-supervised video object segmentation methods are used as baselines, including RANet [36], GMN [22], STM [29], and HMMN [28]. In our experiment, these segmentation methods are fine-tuned multiple times on our generated dataset by their released models and codes, and the best results are reported in this paper. Furthermore, the quantitative results of mask segmentation are reported by two metrics, *i.e.*, intersection over union (IOU) and binary

Table 3. Ablation study of complete network and cycle consistency loss. *MPN* denotes mask prediction network.

	PSNR↑	SSIM↑	E_{warp} ↓	LPIPS↓
OPN+MPN	29.149	0.9013	0.1506	0.4282
OPN+MPN+$\mathcal{L}_c$	30.017	0.9109	0.1490	0.4227
CPVINet+MPN	26.924	0.8696	0.1684	0.8161
CPVINet+MPN+$\mathcal{L}_c$	27.952	0.8765	0.1636	0.8138
E2FGVI+MPN	28.139	0.8857	0.1513	0.5521
E2FGVI+MPN+$\mathcal{L}_c$	29.721	0.8987	0.1501	0.5496
w/o $\mathcal{L}_c$	29.522	0.9001	0.1476	0.4235
final model	30.834	0.9135	0.1452	0.4171

Table 4. Ablation study of mask annotation location.

annotation location	PSNR↑	SSIM↑	E_{warp} ↓	LPIPS↓
first frame	30.834	0.9135	0.1452	0.4171
middle frame	30.826	0.9179	0.1461	0.4196
last frame	30.796	0.9165	0.1443	0.4147

cross entropy (BCE) loss.

Experimental results and analysis. Different from the existing semi-supervised video object segmentation methods, the proposed mask prediction network use the completed current frame and the subsequent frame as input to generate the segmentation masks. Ideally, the difference between these two frames is only the corrupted regions. Tab.2 lists the values of BCE and IOU of our method and the baselines on Youtube-vos [40] and DAVIS [24] datasets. It can be observed that our mask prediction network obtains the best results. Fig.7 shows segmentation results obtained by different methods. It can be seen from Fig.7 that the proposed mask prediction network can obtain the segmentation result closer to ground truth, while the segmentation results of RANet, GMN, STM, and HMMN lose part of the corrupted regions. This also verifies that the inpainting effect of the methods concatenating VOS and VI is unsatisfactory.

4.4. Ablation Studies

Effectiveness of Complete Network. To verify the effectiveness of the proposed frame completion network, we test the completion network trained in a fully-supervised manner, and the testing results are shown in Tab.1. Compared with the fully-supervised baselines, our model obtains the best performance. In addition, we also conduct semi-supervised video inpainting tests using the best three baseline models instead of our complete network. As shown in Tab.3, our semi-supervised model outperforms the other three combined models, demonstrating our completion network's superiority in video inpainting.

Effectiveness of Cycle Consistency. The cycle consistency loss $\mathcal{L}_c$ is used to regularize the training parameters of both the completion network and mask prediction network. In this section, we verify the effectiveness of $\mathcal{L}_c$. As shown in Tab.3, the performance of the four models trained with $\mathcal{L}_c$ is improved compared to the model trained without $\mathcal{L}_c$. Therefore, we can draw the conclusion that $\mathcal{L}_c$ can facilitate the complete network and mask prediction network to generate reliable and consistent correspondences, thereby improving the quality of inpainted videos.

Table 5. Effectiveness of the feature alignment module constructed with different numbers of DCA blocks.

N	0	1	2	3	4	5	6
PSNR↑	26.667	29.569	30.174	30.525	30.834	30.839	30.851
SSIM↑	0.8580	0.9036	0.9081	0.9124	0.9135	0.9137	0.9156
E_{warp} ↓	0.1668	0.1554	0.1511	0.1464	0.1452	0.1453	0.1455
LPIPS↓	0.6916	0.4302	0.4286	0.4213	0.4171	0.4173	0.4172

Table 6. The impact of different numbers of annotated frames in the video on inpainting results.

Annotation	1	2	3	5	7
PSNR↑	30.834	30.918	30.974	31.086	31.056
SSIM↑	0.9135	0.9142	0.9147	0.9156	0.9196
E_{warp} ↓	0.1452	0.1443	0.1437	0.1430	0.1424
LPIPS↓	0.4171	0.4147	0.4124	0.4108	0.4102

Influence of mask annotation location. Notably, in our framework, the annotated mask can be any frame of the video. Tab.4 investigates the effect of the location of the known annotation mask, where we explore three different locations in the video on the inpainting results. As shown in Tab.4, the inpainting result difference caused by the location of the mask is minimal and negligible. This shows that our semi-supervised framework is robust to the annotated locations of masks.

Effectiveness of Feature Alignment module. In Tab.5, we verify the effectiveness of the feature alignment module constructed with different numbers of DCA blocks. We can see that the feature alignment module constructed using DCA blocks can improve the effect of video inpainting, and the more the number of stacked DCA blocks, the better the performance. Considering the time cost, we stack four layers of DCA blocks in the feature alignment module.

Refinement of Inpainting Results. We investigate the impact of annotating different numbers of video frames in the video on inpainting results. As shown in Tab.6, the inpainting quality of our method is further improved with the increases of annotated frames. This means that our method can improve the video inpainting quality by increasing the number of annotated frames in some specific environments.

5. Conclusion

In this paper, we formulate a new task termed semi-supervised video inpainting and propose an end-to-end trainable framework consisting of completion network and mask prediction network to tackle it. This task is essential for real-world applications since it can significantly reduce annotation cost. Experimental results show that the proposed method is effective in semi-supervised video inpainting tasks. Notably, we also tailor a new dataset for the semi-supervised video inpainting task, which can effectively facilitate subsequent research.

References

[1] Ya-Liang Chang, Zhe Yu Liu, Kuan-Ying Lee, and Winston Hsu. Free-form video inpainting with 3d gated convolution and temporal patchgan. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 9066–9075, 2019.

[2] Xi Chen, Zuoxin Li, Ye Yuan, Gang Yu, Jianxin Shen, and Donglian Qi. State-aware tracker for real-time video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9384–9393, 2020.

[3] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 764–773, 2017.

[4] Chen Gao, Ayush Saraf, Jia-Bin Huang, and Johannes Kopf. Flow-edge guided video completion. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 713–729, 2020.

[5] Miguel Granados, James Tompkin, Kwang In Kim, Oliver Grau, Jan Kautz, and Christian Theobalt. How not to be seen-object removal from videos of crowded scenes. *Computer Graphics Forum*, 31(2):219–228, 2012.

[6] Zhang Haotian, Mai Long, Wang Hailin, JinZha ando wen, and Ning Xu; John Collomosse. An internal learning approach to video inpainting. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2720–2729, 2019.

[7] P. Hu, G. Wang, X. Kong, J. Kuen, and Y. Tan. Motion-guided cascaded refinement network for video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1400–1409, 2018.

[8] Jiabin Huang, Sing Bing Kang, Narendra Ahuja, and Johannes Kopf. Temporally coherent completion of dynamic video. *ACM Transactions on Grapics (TOG)*, 35(6):196.1–196.11, 2016.

[9] Xuhua Huang, Jiarui Xu, Yu-Wing Tai, and Chi-Keung Tang. Fast video object segmentation with temporal aggregation network and dynamic template matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8879–8889, 2020.

[10] Jaeyeon Kang, Seoung Wug Oh, and Seon Joo Kim. Error compensation framework for flow-guided video inpainting. In *European Conference on Computer Vision*, pages 375–390, 2022.

[11] Dahun Kim, Sanghyun Woo, Joon-Young Lee, and In So Kweon. Deep blind video decaptioning by temporal aggregation and recurrence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4263–4272, 2019.

[12] Dahun Kim, Sanghyun Woo, Joon-Young Lee, and In So Kweon. Recurrent temporal aggregation framework for deep video inpainting. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 42(5):1038–1052, 2020.

[13] Soo Ye Kim, Kfir Aberman, Nori Kanazawa, Rahul Garg, Neal Wadhwa, Huiwen Chang, Nikhil Karnad, Munchurl Kim, and Orly Liba. Zoom-to-inpaint: Image inpainting with high-frequency details. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 477–487, 2022.

[14] Wei-Sheng Lai, Jia-Bin Huang, Oliver Wang, Eli Shechtman, Ersin Yumer, and Ming-Hsuan Yang. Learning blind video temporal consistency. In *Proceedings of the European conference on computer vision (ECCV)*, pages 179–195, 2018.

[15] Dong Lao, Peihao Zhu, Peter Wonka, and Ganesh Sundaramoorthi. Flow-guided video inpainting with scene templates. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 14599–14608, 2021.

[16] Sungho Lee, Seoung Wug Oh, DaeYeun Won, and Seon Joo Kim. Copy-and-paste networks for deep video inpainting. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 4413–4421, 2019.

[17] Zhen Li, Cheng-Ze Lu, Jianhua Qin, Chun-Le Guo, and Ming-Ming Cheng. Towards an end-to-end framework for flow-guided video inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17562–17571, 2022.

[18] Miao Liao, Feixiang Lu, Dingfu Zhou, Sibo Zhang, Wei Li, and Ruigang Yang. Dvi: Depth guided video inpainting for autonomous driving. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 1–17, 2020.

[19] Rui Liu, Hanming Deng, Yangyi Huang, Xiaoyu Shi, Lewei Lu, Wenxiu Sun, Xiaogang Wang, Jifeng Dai, and Hongsheng Li. Fuseformer: Fusing fine-grained information in transformers for video inpainting. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 14040–14049, 2021.

[20] Ruixin Liu, Bairong Li, and Yuesheng Zhu. Temporal group fusion network for deep video inpainting. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(6):3539–3551, 2022.

[21] Ruixin Liu, Zhenyu Weng, Yuesheng Zhu, and Bairong Li. Temporal adaptive alignment network for deep video inpainting. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 927–933, 2020.

[22] Xiankai Lu, Wenguan Wang, Danelljan Martin, Tianfei Zhou, Jianbing Shen, and Van Gool Luc. Video object segmentation with episodic graph memory networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 661–679, 2020.

[23] Alasdair Newson, Andres Almansa, Matthieu Fradet, Yann Gousseau, and Patrick Perez. Video inpainting of complex scenes. *SIAM Journal on Imaging Sciences*, 7(4):1993–2019, 2014.

[24] 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 (CVPR)*, pages 724–732, 2016.

[25] F. Perazzi, J. Pont-Tuset, B. Mcwilliams, L. Van Gool, 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 (CVPR)*, pages 724–732, 2016.

[26] Jingjing Ren, Qingqing Zheng, Yuanyuan Zhao, Xuemiao Xu, and Chen Li. Dlformer: Discrete latent transformer for video inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3511–3520, 2022.

[27] Andreas Robinson, Felix Jaremo Lawin, Martin Danelljan, Fahad Shahbaz Khan, and Michael Felsberg. Learning fast and robust target models for video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7406–7415, 2020.

[28] Hongje Seong, Seoung Wug Oh, Joon-Young Lee, Seongwon Lee, Suhyeon Lee, and Euntai Kim. Hierarchical memory matching network for video object segmentation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 12889–12898, 2021.

[29] Oh Seoung, Wug, Lee Joon-Young, Xu Ning, and Kim Seon, Joo. Video object segmentation using space-time memory networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 9225–9234, 2019.

[30] Oh Seoung, Wug, Lee Sungho, Lee Joon-Young, and Kim Seon, Joo. Onion-peel networks for deep video completion. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 4402–4411, 2019.

[31] Mingjie Sun, Jimin Xiao, Eng Gee Lim, Bingfeng Zhang, and Yao Zhao. Fast template matching and update for video object tracking and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10791–10799, 2020.

[32] Chuan Wang, Haibin Huang, Xiaoguang Han, and Jue Wang. Video inpainting by jointly learning temporal structure and spatial details. In *Proceedings of the AAAI Conference on Artificial Intellignce (AAAI)*, pages 5232–5239, 2019.

[33] Qiang Wang, Li Zhang, Luca Bertinetto, Weiming Hu, and Philip HS Torr. Fast online object tracking and segmentation: A unifying approach. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1328–1338, 2019.

[34] Yi Wang, Ying-Cong Chen, Xin Tao, and Jiaya Jia. Vc-net: A robust approach to blind image inpainting. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 752–768, 2020.

[35] Yi Wang, Xin Tao, Xiaojuan Qi, Xiaoyong Shen, and Jiaya Jia. Image inpainting via generative multi-column convolutional neural networks. *Advances in neural information processing systems (NIPS)*, 31, 2018.

[36] Ziqin Wang, Jun Xu, Li Liu, Fan Zhu, and Ling Shao. Ranet: Ranking attention network for fast video object segmentation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[37] Zhiliang Wu, Changchang Sun, Hanyu Xuan, Kang Zhang, and Yan Yan. Divide-and-conquer completion network for video inpainting. *IEEE Transactions on Circuits and Systems for Video Technology*, Early Access, 2022.

[38] Zhiliang Wu, Kang Zhang, Hanyu Xuan, Jian Yang, and Yan Yan. Dapc-net: Deformable alignment and pyramid context completion networks for video inpainting. *IEEE Signal Processing Letters*, 28:1145–1149, 2021.

[39] Haozhe Xie, Hongxun Yao, Shangchen Zhou, Shengping Zhang, and Wenxiu Sun. Efficient regional memory network for video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1286–1295, 2021.

[40] Ning Xu, Linjie Yang, Yuchen Fan, Jianchao Yang, Dingcheng Yue, Yuchen Liang, Brian Price, Scott Cohen, and Thomas Huang. Youtube-vos: Sequence-to-sequence video object segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 603–619, 2018.

[41] Rui Xu, Xiaoxiao Li, Bolei Zhou, and Chen Change Loy. Deep flow-guided video inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3723–3732, 2019.

[42] Yingchen Yu, Fangneng Zhan, Shijian Lu, Jianxiong Pan, Feiying Ma, Xuansong Xie, and Chunyan Miao. Wavefill: A wavelet-based generation network for image inpainting. In *IEEE International Conference on Computer Vision (ICCV)*, pages 14094–14103, 2021.

[43] X. Zeng, R. Liao, L. Gu, Y. Xiong, S. Fidler, and R. Urtasun. Dmm-net: Differentiable mask-matching network for video object segmentation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 3928–3937, 2019.

[44] Yanhong Zeng, Jianlong Fu, and Hongyang Chao. Learning joint spatial-temporal transformations for video inpainting. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3723–3732, 2020.

[45] Yanhong Zeng, Jianlong Fu, Hongyang Chao, and Baining Guo. Learning pyramid-context encoder network for high-quality image inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1486–1494, 2019.

[46] Kaidong Zhang, Jingjing Fu, and Dong Liu. Flow-guided transformer for video inpainting. *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.

[47] Kaidong Zhang, Jingjing Fu, and Dong Liu. Inertia-guided flow completion and style fusion for video inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5982–5991, 2022.

[48] Lu Zhang, Zhe Lin, Jianming Zhang, Huchuan Lu, and You He. Fast video object segmentation via dynamic targeting network. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 5582–5591, 2019.

[49] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 586–595, 2018.

[50] Xueyan Zou, Linjie Yang, Ding Liu, and Yong Jae Lee. Progressive temporal feature alignment network for video inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16448–16457, 2021.