frontiers | Frontiers in Neurorobotics

TYPE Methods
PUBLISHED 13 October 2022
DOI 10.3389/fnbot.2022.1002453

Check for updates

OPEN ACCESS

EDITED BY
Chang Li,
Hefei University of Technology, China

REVIEWED BY
Kei Majima,
National Institutes for Quantum and Radiological Science and Technology, Japan
Zhang Zhongzhen,
Hefei University of Technology, China
Li Jing,
Central University of Finance and Economics, China

*CORRESPONDENCE
Tao Xue
xuetao@xpu.edu.cn

RECEIVED 25 July 2022
ACCEPTED 20 September 2022
PUBLISHED 13 October 2022

CITATION
Ma P and Xue T (2022) Learning a spatial-temporal texture transformer network for video inpainting. *Front. Neurorobot.* 16:1002453. doi: 10.3389/fnbot.2022.1002453

Learning a spatial-temporal texture transformer network for video inpainting

Pengsen Ma and Tao Xue*

School of Computer Science, Xi'an Polytechnic University, Xi'an, China

We study video inpainting, which aims to recover realistic textures from damaged frames. Recent progress has been made by taking other frames as references so that relevant textures can be transferred to damaged frames. However, existing video inpainting approaches neglect the ability of the model to extract information and reconstruct the content, resulting in the inability to reconstruct the textures that should be transferred accurately. In this paper, we propose a novel and effective spatial-temporal texture transformer network (STTTN) for video inpainting. STTTN consists of six closely related modules optimized for video inpainting tasks: feature similarity measure for more accurate frame pre-repair, an encoder with strong information extraction ability, embedding module for finding a correlation, coarse low-frequency feature transfer, refinement high-frequency feature transfer, and decoder with accurate content reconstruction ability. Such a design encourages joint feature learning across the input and reference frames. To demonstrate the advancedness and effectiveness of the proposed model, we conduct comprehensive ablation learning and qualitative and quantitative experiments on multiple datasets by using standard stationary masks and more realistic moving object masks. The excellent experimental results demonstrate the authenticity and reliability of the STTTN.

KEYWORDS

transformer, video inpainting, texture converter, spatial-temporal, deep learning

1. Introduction

Video inpainting involves smearing moving or stationary objects in a video frame sequence using masks. The smeared parts are filled back based on the current frame and the content information of other frames of the video, and the repaired video should have the effect that the smeared positions 'disappear'. Typical applications are video restoration (Kim et al., 2018; Chang et al., 2019a,b), watermark removal (Zou et al., 2021), object removal (Perazzi et al., 2016; Chang et al., 2019d), etc. The closer the smeared area is to the actual video after being repaired, the better the repair effect.

Video inpainting needs to combine time domain and spatial domain information to process video frames. The spatial information in the current frame is searched, followed by the appropriate frames in other frames as reference frames to search the time domain information. Finally, the two parts of information are integrated and filled back into the original frame to complete the repair of the mask position (Zeng et al., 2020; Liu et al., 2021a). Video inpainting should first consider whether the missing information of the current frame is 'exposed' in other frames. If the missing information of the current frame is found in other frames, then the current frame should be used as a reference frame. Valuable features should be matched, extracted, and transmitted to the input frame as information to repair the mask position. Although the recent appearance of deep learning has made significant progress in the image and video inpainting (Iizuka et al., 2017; Boßmann et al., 2019; Yu et al., 2019), a model's ability to capture useful information and reconstruct it in video frames is still fragile (Chang et al., 2019a; Lee et al., 2019; Oh et al., 2019; Xu et al., 2019).

In summary, video inpainting needs to integrate the information acquired in time and space and effectively transform and fill it back into the restored image. The more complicated and challenging portion of this process is 2-fold: What information (in time) should be extracted from the reference frame? How can the information of the reference frame and the current frame be effectively extracted and used (spatially)?

We put forward the spatial-temporal texture transformer network to solve these problems in video inpainting. STTTN is divided into the following six parts: (1) Make the feature similarity measure more accurate and frame pre-repair. (2) By introducing Regional Normalization (RN) (Yu et al., 2020), spatial pixels are divided into different regions according to a mask, which solves the problem of the deviation of the mean and variance, thus constructing an encode with stronger information extraction ability. (3) To embed the information in the image, similar to the standard transformer structure (Dosovitskiy et al., 2020; Liu et al., 2021c), a related texture information embedding module (RE) is introduced to embed the reference and input frames. (4) Coarse low-frequency feature transfer (CLFT) is used to convert low-frequency information such as contours from reference frames to input frames. (5) Refinement high-frequency feature transfer (RHFT) is used to transfer further delicate texture information such as image details to the input frame and perfect the repair mask. (6) Information such as the feature texture, which is composed of various pieces of information, is obtained. Similar to the encoder, we added learnable region normalization in the decoder to help the fusion of corrupted and uncorrupted areas and more stable modified video frames.

These six parts help each other build a powerful space-time transformer video restoration network. There are three steps in the inpainting: the input frame and reference frames before the encoder are pre-repaired, the coarse low-frequency texture features are transferred and repaired, and finally the high-frequency texture features related to details are further refined and repaired. The three parts are paved from low to high, and the complete repair process is formed layer by layer.

To verify the progressive nature of the STTTN, we carried out a large number of qualitative and quantitative experiments. An ablation study of four parts of the texture converter and loss function proved the effectiveness of each part of the components.

Our research makes the following contributions:

- A novel and effective spatio-temporal texture converter for video restoration, which achieves significant improvements over the state-of-the-art approaches, is proposed.
- Regional normalization (RN) introduced into the encoder and decoder creatively stabilizes the effect of video restoration.
- From the results of various mask experiments on multiple datasets, STTTN achieves excellent results visually and in terms of evaluation parameters.

2. Related studies

Video Inpainting refers to smearing some fixed areas or moving objects in a video and filling the smeared areas back in a generated way. It is required as far as possible to leave no trace (that is, it is not easy to detect by the naked eye). Image inpainting fills the missing area in a single image. In a narrow sense, image inpainting is a subset of video inpainting. The difference between them is that video inpainting needs to integrate the temporal and spatial information acquired from all video frames and effectively transform and fill it back into the restored image. Image inpainting only needs to make use of the spatial information outside the mask of the current picture without considering the information of other frames because it is only a single picture without any temporal information. From the perspective of repair methods, video inpainting is mainly divided into two methods. One is explicit repair, that is, it acts directly on the image and repairs from the pixel level. The other is implicit inpainting, that is, it acts on image coding and inpainting from the feature level.

2.1. Explicit inpainting

Explicit inpainting is a network design pattern of 'graph to graph' and is directly constructed. Before the popularity of deep learning, the main image and video inpainting methods were diffusion-based and patch-based methods. The central idea of diffusion-based methods (Ballester et al., 2001; Levin et al., 2003) was to predict and fill holes according to the pixels around the region to be filled. For example, the fast marching method (FFM) (Telea, 2004), and the fluid dynamics method (FDM), are classic diffusion-based inpainting algorithms, but their limitations are also pronounced. It is more suitable for image pinhole inpainting with little color change and a simple scene. Patch-based methods complete the repair task by finding the most appropriate area to fill the hole and pasting it into the place to be inpainting. For a single image, it searches for the most suitable area outside the hole area, that is, spatial-based inpainting. Video inpainting considers all regions in the current frame and other frames to select the most appropriate area to fill, which is a temporal-based and spatial-based approach. Generally speaking, both diffusion-based and patch-based methods are predicted or pasted to the area to be inpainting according to the area around the hole. They can not capture advanced semantic information. They are repaired in a way that can't be learned, so they are also called Non-learning-based inpainting.

The appearance of deep learning makes up for the deficiency of content restoration in complex and dynamic motion areas from multiple objects. At present, the primary display inpainting methods based on deep learning are divided into the following two types. The first is optical flow calculation, that is, the 'movement trend' of pixels is calculated based on the difference between the front frame and the back frame. This 'trend' is used to predict the color propagation to fill the missing mask block (Xu et al., 2019; Lao et al., 2021). The second is 3D-CNN/RNN, which directly stacks the frames in time series according to the number of channels to form a large matrix for convolution calculation (Kim et al., 2019; Wang et al., 2019). This method is cumbersome, takes up considerable video memory, and consumes many computing resources. In addition, the effect worsens when encountering some frequently switched and complex video scenes, so there is little room for improvement.

2.2. Implicit inpainting

Traditional restoration methods often consume too much memory and lead to a long reasoning time, and can not effectively capture texture information in the temporal domain and spatial domain. At the same time, a network based on implicit inpainting is small and exquisite, with a relatively strong effect and ample room for improvement. The depth representation of the image is obtained using an encoder. Then, a series of patching operations are performed on the image representation: attention (Tang et al., 2019; Liu et al., 2021a; Shu et al., 2021; Zou et al., 2021), generative adversarial network (Chang et al., 2019a,d; Zou et al., 2021), gated convolution (Yu et al., 2019), region normalization (Suin et al., 2021), etc. This is mapped (decoder) back to the image to generate video frames after patching is completed.

Video inpainting needs to find high-level semantic information in time and space; that is, it needs to capture long-distance dependencies. In recent years, the appearance of the transformer (Arnab et al., 2021; Chen et al., 2021; Wang and Wang, 2022) has provided a new solution for vision tasks. Compared with traditional CNN (Gu et al., 2022) and RNN-based (Lin et al., 2022) methods, transformers have better capability to understand shape and geometry and capture the dependencies between long distances. We propose a spatial-temporal texture transformer network (Han et al., 2020). We learn the feature relationship between video frames and within frames according to the semantic consistency of context to complete hole filling, which is effective and efficient for video inpainting.

3. Approach

First, we introduce the overall architecture design of STTTN and then explain its five essential components in detail, namely encoder, relevance embedding (RE), coarse low-frequency feature transfer (CLFT), refining high-frequency feature transfer (RHFT), and decoder. Finally, we combed the whole video restoration process and explained the model's loss function more clearly through pseudo-code.

3.1. Overall design

We use implicit inpainting to repair the video from the feature level. To determine the defects of the previous architecture and achieve a better inpainting effect, we erase the time domain search part of the previous baseline model (Zeng et al., 2020; Liu et al., 2021a,b) and find that the inpainting ability is significantly reduced and even lags behind the effect of many nonspatiotemporal video inpainting models. This shows that most of the previous study explored how to search the memory in the temporal domain but neglected to examine the depth representation of the obtained images; that is, only by introducing temporal domain search can the model perform better, but their spatial domain search is not as good as the original image inpainting model, which needs stronger information extraction ability and fine content reconstruction ability. Therefore, we built a new encoder and decoder architecture, which makes STTTN have a more vital ability to capture image structure and information in the time domain. The overall architecture idea is that the depth representation of the image is obtained by the encoder. Then, after the image representation is repaired, the image is mapped back to the decoder to generate the repaired image frame.

The overall structure of the spatial-temporal texture transformer is shown in Figure 1. This structure contains six parts. The first part is the preprocessing of video frames before they are input into the network. The input framesΔ and reference framesΔ represent the pre-repaired input frames and reference frames, respectively. Precisely, we mask at the position where the reference frames are consistent with input

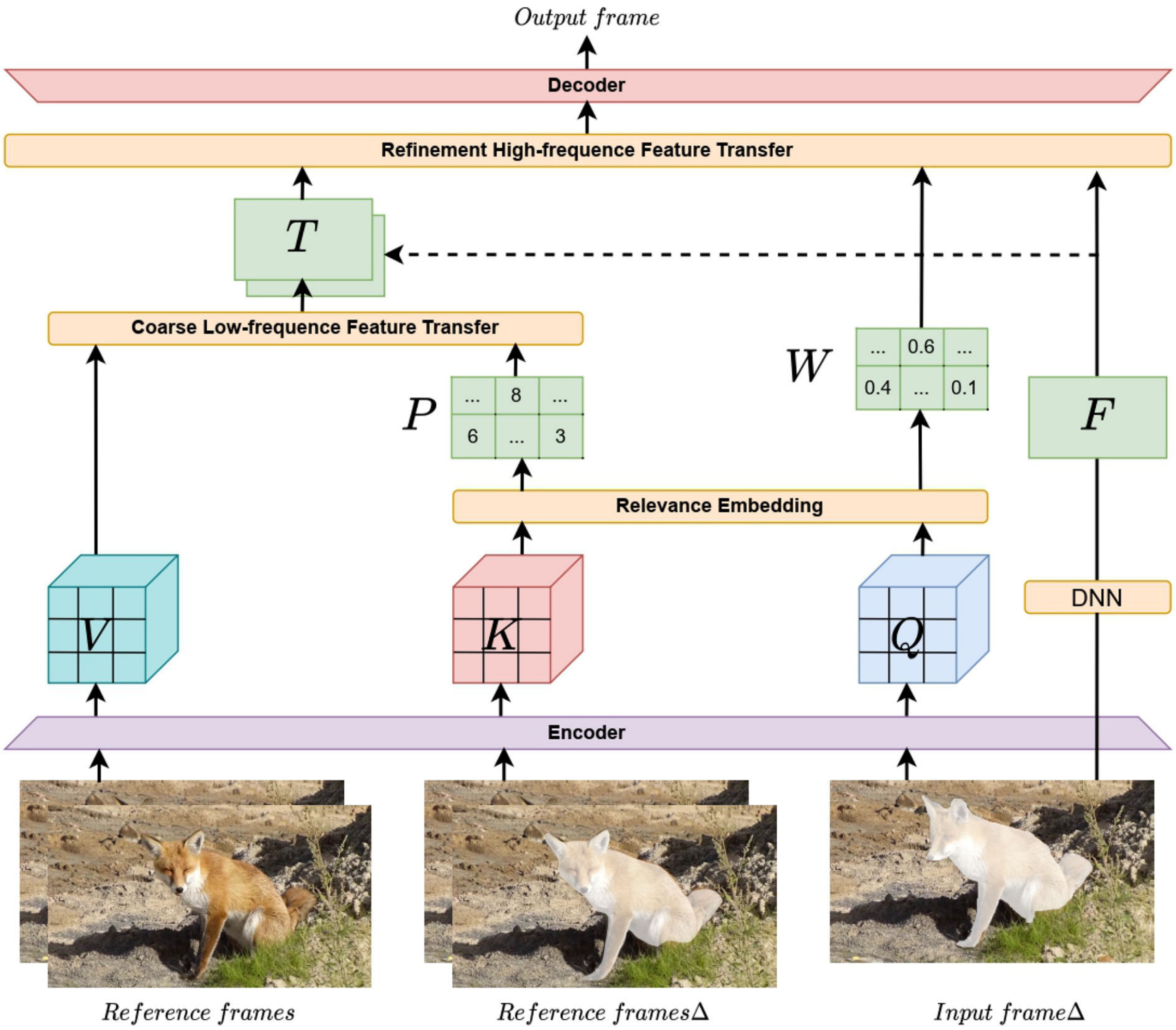

FIGURE 1
Overview of spatial-temporal texture transformer network (STTTN) structure.

frames and prerepair the mask to obtain reference frames∆ to ensure domain consistency with input frames∆. Here, we fix the random seed to ensure that the mask position and size of the input frame and the reference frame are consistent, and then we randomly switch the random seed to generate a new mask when processing the next frame. It is more accurate to measure the feature similarity with the domain (Yang et al., 2020), in which the prerepair method is a fast marching method (FFM) (Telea, 2004). The remaining five parts are an encoder with super information extraction ability, embedding module (RE) to find a correlation, coarse low-frequency feature transfer (CLFT), refinement of high-frequency feature transfer (RHFT), and accurate content decoder with reconstruction capability. Details are discussed below.

3.2. Spatial-temporal texture transformer

3.2.1. Encoder

The traditional video processing and image inpainting methods use feature normalization (FN) to help with network training (Kobla et al., 1996; Wang et al., 2020), but they are often performed on the entire frame without considering the impact of pixels in the corrupted region on the mean/variance. By introducing regional normalization (RN), the spatial pixels are divided into different regions according to the mask, and then the mean and variance are calculated in different regions. As shown in Figure 2, we embed basic regional normalization (RN-B) on the encoder, which normalizes the corrupted and uncorrupted regions based on the input mask. This allows mean and variance offsets to be more accurate, which is more

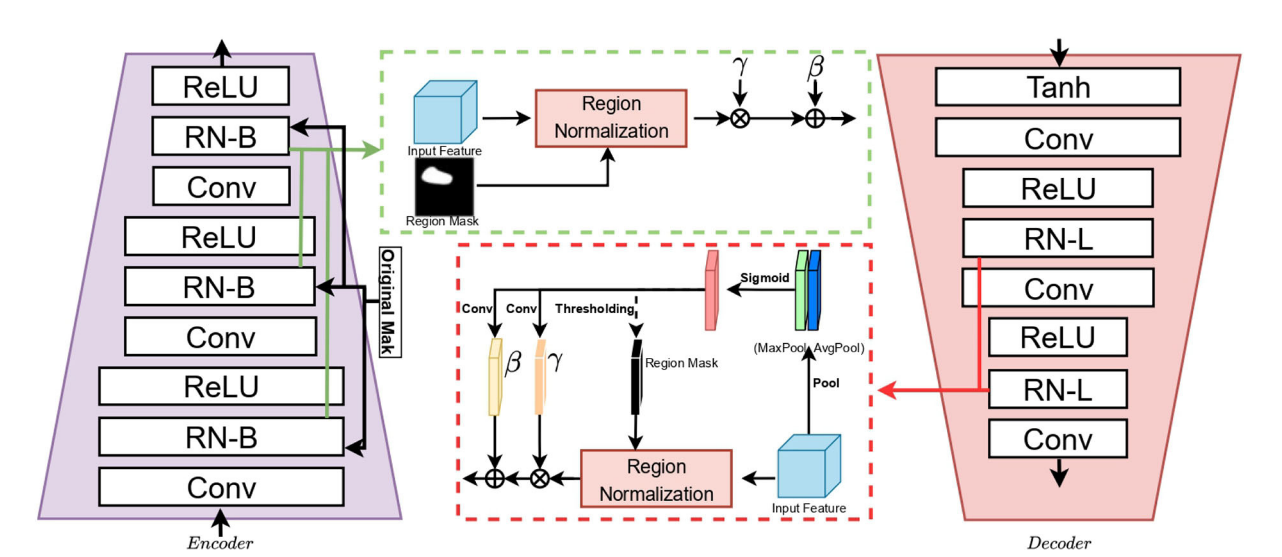

FIGURE 2
Structure of encoder and decoder.

conducive to obtaining a deep image representation, which can more comprehensively extract helpful information from video frames.

The input to the encoder network consists of three frames (input frames∆, reference frames∆, and reference frames). Inp, Inp∆, Ref, and Ref∆ represent the input frames, input frames∆, reference frames, and reference frames∆, respectively. Inp∆ and Ref∆ consist of an RGB image, hole mask, and no-hole mask. The hole mask on the RGB image is a single-channel greyscale image, and the no-hole mask is an area other than the hole mask area. These inputs are concatenated along the channel axis to form a 5-channel image before being fed into the first layer. Ref are composed of three-channel RGB images. Given an input feature $F \in R_{C\times H\times W}$ and binary region mask $M \in R_{1\times H\times W}$ that indicates a corrupted region, for each channel, there are two sets of learnable parameters γ and β for the affine transformation of each region. *Via* the encoder, the obtained image representation consists of three parts: Q (representative attention information), K (attention information of memory frame), and V (representative content information of memory frame).

3.2.2. Relevance embedding

Different from the previous operation of obtaining q, k, v through a linear transformer and then calculating attention (Lee et al., 2021), we obtain Q, K, V with sufficient texture feature information through the encoder, which makes it easier to find the correlation between the input frame and the reference frames in the time domain. First, we use RE to estimate the similarity between Q and K, so as to establish the correlation between Inf and Ref.We unfold Q and K into patches, denoted as $q_i\left(i \in \left[1, H_{Inp} \times W_{Inp}\right]\right)$ and $k_j\left(j \in \left[1, H_{Ref} \times W_{Ref}\right]\right)$. We calculate their similarity by dot multiplying q and k^T, where T represents the transpose operation:

$$R_{i,j} = q_i \times k_j^T \tag{1}$$

The larger $R_{i,j}$, the stronger the correlation between the two feature blocks, and the more texture information can be migrated, and vice versa.

With the correlation $R_{i,j}$ obtained by RE, we can obtain two parts P and W for coarse low-frequency feature transfer and refinement of high-frequency feature transfer, respectively. The specific calculation details are in Sections 3.2.3, 3.2.4.

3.2.3. Coarse low-frequency feature transfer

To transfer the low-frequency information of images, such as contours from the reference frame to the input frame, we designed the coarse low-frequency feature transfer (CLFT). The previous attention mechanism converted $R_{i,j}$ through softmax into a weight directly and then multiplied the weight by V, which is a weighted average of V:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2}$$

However, doing so may transfer a large number of textures that are not useful for the input frame to the target frame, resulting in blurring of the repaired area. To improve the ability to transfer low-frequency texture features for all reference frames. we will correlate the coarse low-frequency features in V

over different temporal domains with input frames *via* CLET. More specifically, we first calculate a coarse low-frequency feature transfer map P in which the i-th element $P_i(i \in [1, H \times W])$ is calculated from the relevance $R_{i,j}$:

$$P_i = \arg\max r_{i,j} \tag{3}$$

That is, each value p_i in map P represents the most relevant position index of a frame on all reference frames with the i-th position of the input frame. The specific calculation process obtains the index corresponding to the maximum value through the second item of the return value of the torch.max() function. After obtaining the most relevant position index, we extract the low-frequency texture features that should be transferred most, so we only need to take the position of the frame that needs to be transferred in the patch v of the unfolded, then we can get the texture feature map T, where each position of T contains the high-frequency texture features of the most similar position in the Ref, where t_i represents the value of the i-th position of T:

$$t_i = v_{p_i} \tag{4}$$

We obtain a rough feature representation T for input frames, which is then used in our refinement of high-frequency feature transfer (RHFT).

3.2.4. Refinement high-frequency feature transfer

High-frequency detail information is also essential for video inpainting (Bishop et al., 2003), so we designed a refinement of high-frequency feature transfer (RHFT). To fuse the most suitable high-frequency texture in the temporal and spatial domains with the input frame, a weight matrix W is calculated from $R_{i,j}$ to represent the confidence of the transferred texture features for each position in T. The specific calculation process for obtaining W is to obtain the maximum value of $R_{i,j}$ through the first item of the return value of the torch.max() function, where W records the specific correlation of the most relevant feature block.

$$W_i = \max R_{i,j} \tag{5}$$

To make full use of the original image information of the input frame, we divide the features of each level into two steps. First, the low-frequency texture features T of multiple frames in the temporal domain are obtained by CLFT, and the feature of the input frame is fusion. The product is then multiplied by the weight matrix W. At this time, W is equivalent to a weighted average of the features, which can more accurately transfer the texture features of the reference frame. Only two feature transfers cannot modify the input frame well, so we extract the features of the input frame again (Only two feature transfers cannot modify the input frame well, so we extract the features of the input frame again (feature F extracted by the DNN, which is a deep neural network composed of convolution and residual connections of many layers with convolution kernel of 3*3, and stride and padding are 1) and fuse them with high- and low-frequency features. The above operations can be expressed as the following formula:

$$F_{\text{out}} = F + \text{Conv}(\text{Concat}(T, F)) \odot \text{W} \tag{6}$$

Conv, Concat, and $\odot$ represent the convolution (the convolution operation adopted here is consistent with the convolution operation adopted by the DNN above), the concatenation operation, and the dot product, respectively, and Fout is the feature of the spatiotemporal texture output of the input frame combined with reference frames.

3.2.5. Decoder

In the deep network, each corrupted area and uncorrupted area are increasingly difficult to distinguish, and the corresponding mask is difficult to obtain (Yu et al., 2020). To enhance the reconstruction ability of the image, we insert the learnable RN (RN-L) into the decoder to automatically detect the mask and nonmask. Regions are individually normalized, and a global affine transformation is performed to enhance their fusion. Finally, the repaired video frame is obtained by outputting the newly repaired representation through the decoder.

In summary, the STTTN can effectively transfer relevant high- and low-frequency texture features from the reference frames into the input frame, producing a more accurate mask filling process. For a more precise illustration of how we perform video inpainting, as in Algorithm 1, we detail the pseudocode of our entire calculation process:

```
Input: video $X$, hole $H$, validity $V$
Output: complection video $Y$
for $l$ in reference frame indices do
    $v_l = \text{FFM}(X_l, H_l)$
    $v_l = \text{Encoder}(X_l)$
    $k_l = \text{Encoder}(\text{FFM}(X_l, H_l), V_l)$
end
for $i$ in target frame indices do
    $q_i = \text{Encoder}(\text{FFM}(X_i, H_i), V_i)$
    $P, W = \text{RE}(q_i, k_l)$
    $T = \text{CLFT}(P, v_l)$
    $F = \text{DNN}(\text{FFM}(x_i))$
    $F_{out} = F + \text{Conv}(\text{Concat}(T, F)) \odot \text{W}$
    $Y_i = \text{Decoder}(F_{out})$
end
```

Algorithm 1. Spatial-temporal texture transformer network for video inpainting.

3.3. Loss function

The total loss consists of the following two components:

$$\mathcal{L}_{\text{overall}} = \mathcal{L}_{cha} + 0.01\mathcal{L}_{\text{per}} \tag{7}$$

3.3.1. Charbonnier loss

We abandon the L1 and L2 loss functions because they both cause the image to be too smooth and lack a sense of realism. Instead, we use a more stable loss function: the Charbonnier loss function (Lai et al., 2018a). It can be formulated as

$$\mathcal{L}_{\text{cha}} = \sqrt{\left\|\mathcal{I}^{Out} - \mathcal{I}^{Input}\right\|^2 + \epsilon^2} \tag{8}$$

where Input means original video, Out means synthesized video, and $\epsilon = 10^{-3}$ is a constant to avoid gradient disappearance and explosion.

3.3.2. Perceptual loss

To make more effective use of texture features transferred from reference video, make the inpainted video frames more realistic, and maintain content invariance (Yang et al., 2020), we construct a perceptual loss that consists of two parts:

$$\begin{aligned}\mathcal{L}_{\text{per}} = &\frac{1}{C_i H_i W_i}\left\|\phi_i^{vgg}\left(I^{Out}\right) - \phi_i^{vgg}\left(I^{Ref}\right)\right\|_2^2 + \\ &\frac{1}{C_j H_j W_j}\left\|\phi_j^{Enc}\left(I^{Out}\right) - T\right\|_2^2\end{aligned} \tag{9}$$

The first part is no different from ordinary perceptual loss (Johnson et al., 2016). ϕ_i^{vgg} represents the feature map of the i-th layer of VGG-16 pretrained on ImageNet (Deng et al., 2009); (C_j, H_j, W_j) represents the number of channels, height, and width of the feature map of this layer; and I^{Ref} is the reference video for all frames. T is the texture feature transferred from V in Figure 2.

4. Experiments

4.1. Datasets and evaluation metrics

For a fair comparison of STTTN and other video inpainting models such as previous state-of-the-art versions, we use YouTube-VOS (Xu et al., 2018) and DAVIS (Caelles et al., 2018) as our datasets. The train/validation/test split is consistent with the original split. There are 3471, 474, and 508 video clips, respectively. For DAVIS, we divided its 150 video clips into 90 training sets and 60 validation sets and then randomly selected 30 as test sets.

To test the ability of the model to cope with a variety of practical application scenarios, we use two mask test models, namely, stationary masks and dynamic masks. The static mask means that the position of the fixed mask does not change, and the dynamic mask means that a moving object as a mask forces the mask keep the position transformation in each frame.

Various evaluation criteria are prerequisites to ensure the superior performance of the model. We use PSNR, SSIM, flow warping error (Lai et al., 2018b), video-based Fr'echet inception distance (VFID) (Wang et al., 2018), floating-point operations (FLOPs), and frames per second (FPS) as our evaluation metrics. VFID transfers the FID evaluation from the image to the video task, and the flow warping error measures the temporal stability of a video between the repaired frame and the original frame. FLOPs and FPS test the computing resources required by the model and the fluency of the repaired video, respectively.

4.2. Evaluation

To test the STTTN more comprehensively, we conduct qualitative and quantitative evaluations with five current SOTA methods: VINet (Kim et al., 2019), DFVI (Xu et al., 2019), LGTSM (Chang et al., 2019c), CAP (Lee et al., 2019), STTN (Zeng et al., 2020), and FGVC (Gao et al., 2020).

TABLE 1 Quantitative results of video completion on YouTube-VOS and DAVIS datasets.

	Accuracy								Efficiency	
	YouTube-VOS				DAVIS					
Models	PSNR↑	SSIM↑	VFID↓	E$_{warp}$ ↓	PSNR↑	SSIM↑	VFID↓	E$_{warp}$ ↓	FLOPs↓	FPS↑
VINet	29.20	0.9434	0.072	0.1490	28.96	0.9411	0.199	0.1785	-	-
DFVI	29.16	0.9429	0.066	0.1509	28.81	0.9404	0.187	0.1880	-	-
LGTSM	29.74	0.9504	0.070	0.1859	28.57	0.9409	0.170	0.2566	261B	18.7
CAP	31.58	0.9607	0.071	0.1470	30.28	0.9521	0.182	0.1824	**211B**	15.0
STTN	32.34	**0.9655**	0.053	0.1451	30.67	0.9560	**0.149**	0.1779	233B	24.3
FGVC	31.28	0.9502	-	-	-	-	-	-	-	-
Proposed	**32.68**	0.9654	**0.051**	**0.1421**	**31.32**	**0.9620**	**0.149**	**0.1738**	254B	**36.8**

The bold red font represents the best score, and the blue font represents the second best score.

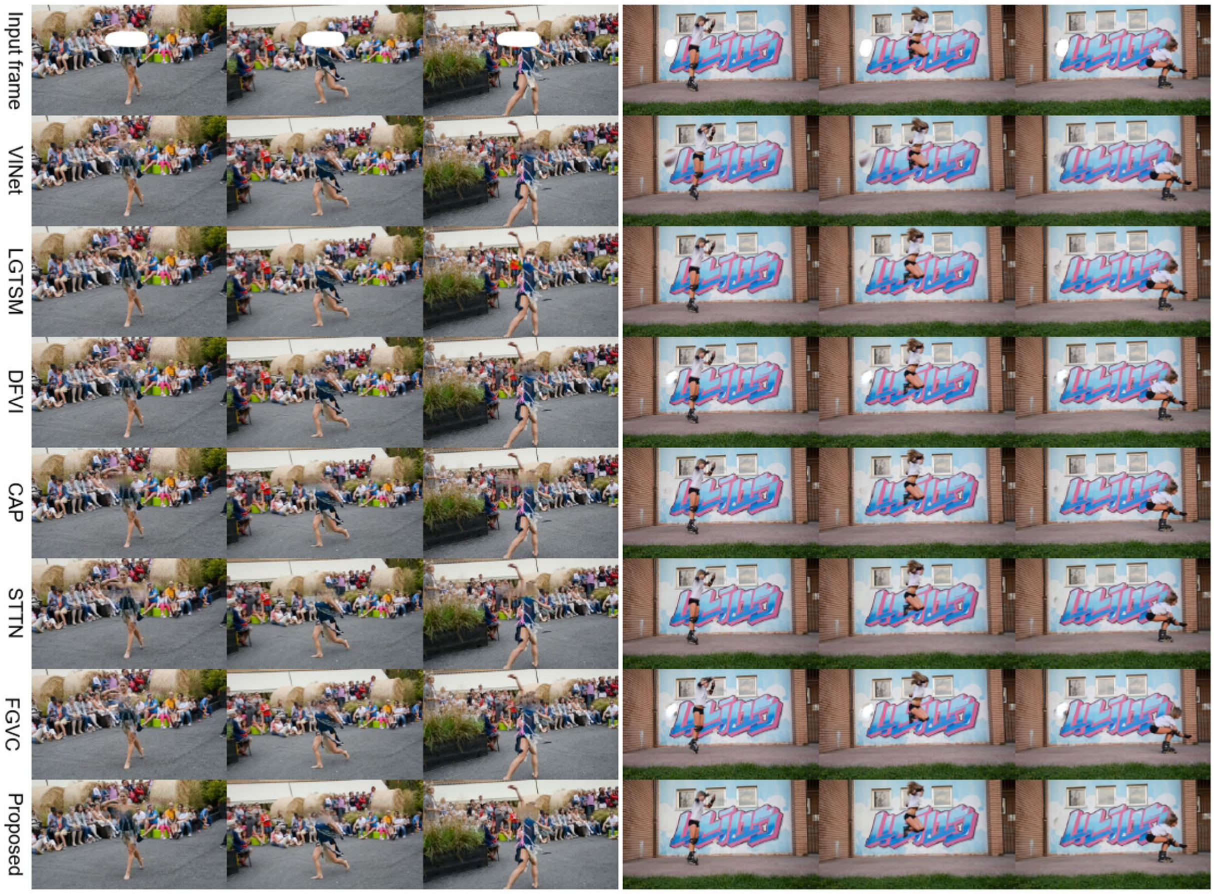

FIGURE 3
Qualitative comparison with other methods for stationary masks.

4.2.1. Quantitative evaluation

As shown in Table 1, on the YouTube-VOS and DAVIS test sets, our proposed STTTN is generally at the highest level, and relatively few FLOPs and higher FPS ensure a lightweight model and the smoothness of the video.

4.2.2. Qualitative evaluation

To demonstrate the effectiveness and generalization of STTTN, as shown in Figures 3, 4, we conduct experiments on dynamic masks and static masks. It can be seen that compared with other current state-of-the-art models, regardless of whether it is a complex or straightforward scene, STTTN achieves the best restoration effect in terms of overall feeling and local details.

4.2.3. User study

To eliminate the tendency of individuals to subjectively use a specific model, we selected 100 students in the school to conduct a user survey and gave each student 12 photos (a total of 7 comparison models, and each model selected two video restoration examples for the testing set). The students chose one image from the two repaired images containing STTTN each time they thought the repair was better and better; that is, each person made ten choices, for a total of $100 \times 12 = 1,200$ voting choices. In Figure 5, the vertical axis represents the percentage for which they believe STTTN repair is better than the current model. The table shows that STTTN consistently outperforms other models compared with other inpainting effects.

4.3. Ablation studies

To verify the effectiveness of each part of STTTN, we carried out ablation learning, which is each part of texture transformation, and loss function, where each part of texture transformation contains four sets of ablation learning and the loss function contains two sets of ablation learning.

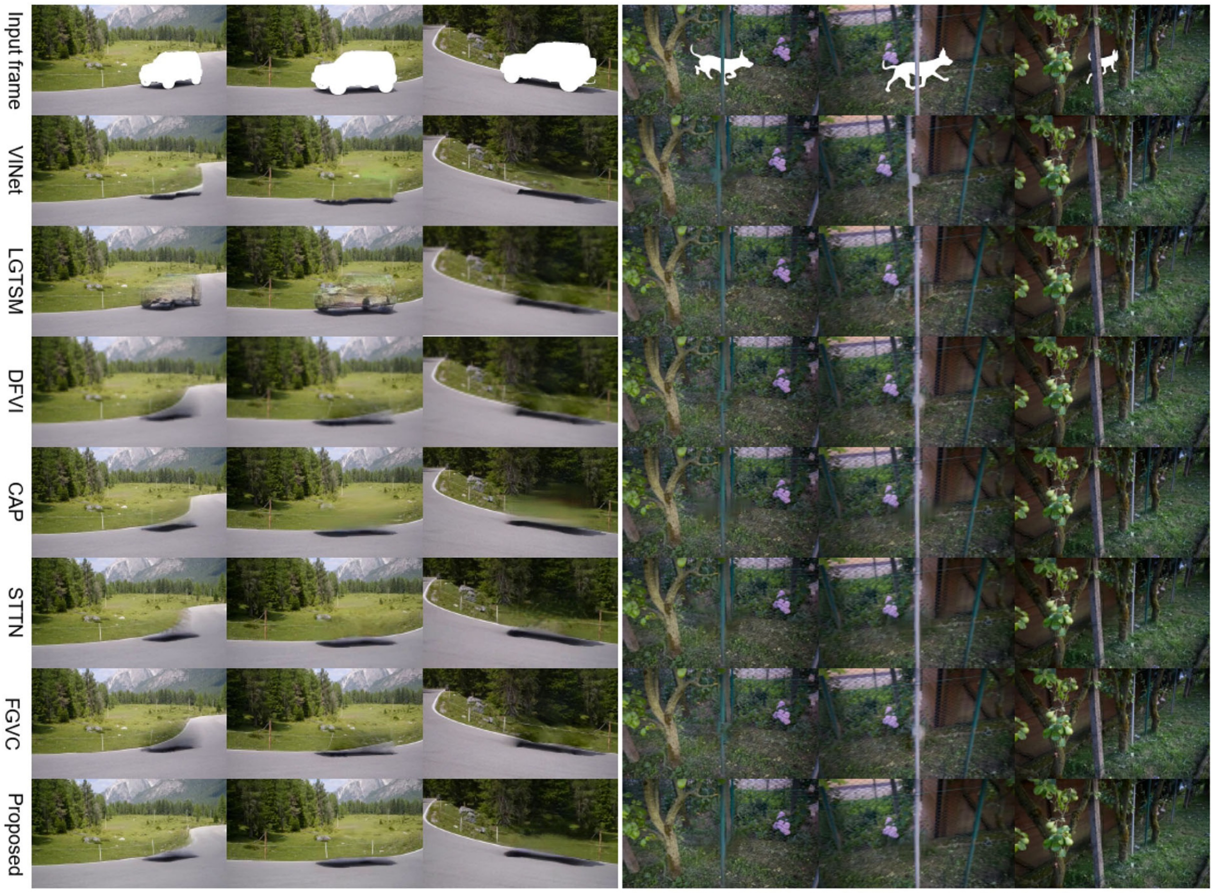

FIGURE 4
Qualitative comparison with other methods for dynamic masks.

4.3.1. Effects of various parts of texture transfer

As shown in Table 2, the texture transfer part is divided into four parts for ablation learning: CLFT, RHFT, encoder, and video frame prerepair. Base means removing these four parts and using the transformer for video repair (similar to a simplified version of STTN). We gradually increase these four parts each time to see the performance of STTTN. When CLFT and RHFT are added to the base, the PSNR increases by 0.25 and 0.19, respectively, so that STTTN can accurately convert the coarse low-frequency feature and refine the high-frequency feature and replace the finely designed encoder with ordinary Q, K, and V extraction. The improvement in PSNR is the most obvious (0.33), indicating that this part enables the texture converter to have more vital information extraction ability and fine content reconstruction ability. To explore whether it is helpful to pre-inpaint the reference frame and the input frame, we put them in the same domain, and the PSNR has a slight improvement of 0.03. In addition to the ablation learning of the four parts that make up the STTTN, we also added experiments to the experiment with regional normalization (RN) in the encoder and decoder to judge the impact of model performance. After adding RN, PSNR and SSIM were improved by 0.87 and 0.012, demonstrating the effectiveness of RN. The above ablation learning demonstrates the importance and effectiveness of the four parts, which complement each other and constitute a powerful texture transformer.

4.3.2. Effects of charbonnier loss and transferal perceptual loss

The five columns in the first row of Figure 6 represent the input frame with mask, use only the L1 loss function, replace the L1 loss with Charbonnier loss (C loss), add the first part of Perceptual Loss (P loss) based on the third column 1) based on column 4, and add the repair effect diagram of Part 2 (P loss 2) of Perceptual Loss. The second row is the uncropped inpainted frames for the four cases. Combining Figure 6 and Table 3, we can see that as we gradually complete the loss function, the

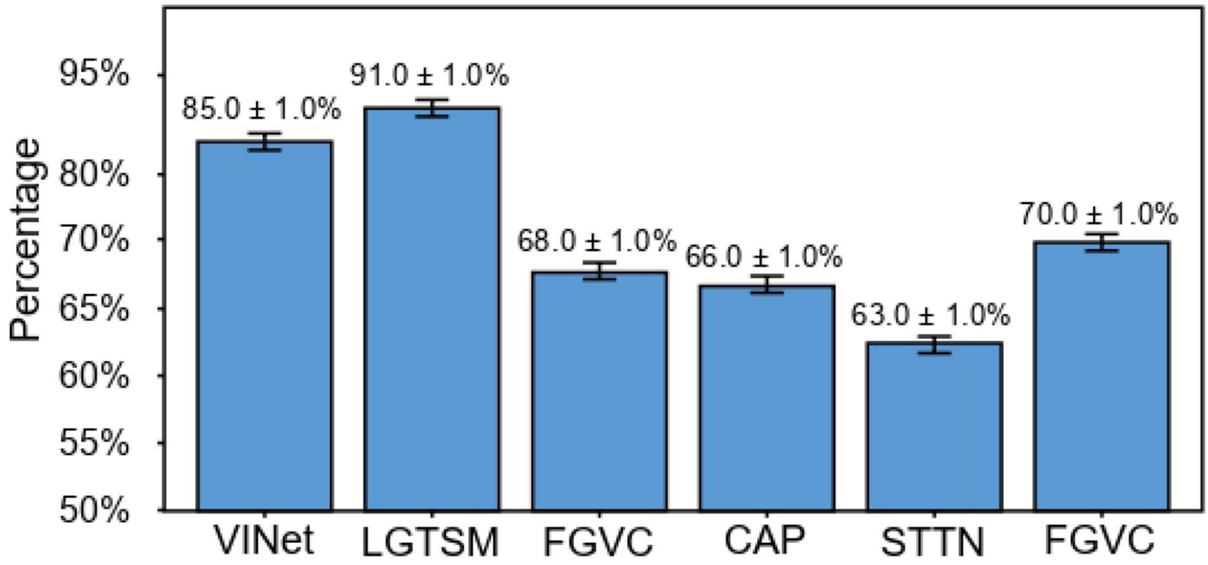

FIGURE 5
User study results for dynamic masks.

TABLE 2 Effects of various parts of texture transfer.

Method	CLFT	RHFT	Encoder	Δ	RN	PSNR↑/SSIM↑/VFID↓/E_{warp} ↓
Base					✓	30.52 / 0.9531 / 0.168 / 0.1810
Base+CLFT	✓				✓	30.77 / 0.9561 / 0.163 / 0.1793
Base+CLFT+RHFT	✓	✓			✓	30.96 / 0.9580 / 0.158 / 0.1774
Base+CLFT+RHFT+Encoder	✓	✓	✓		✓	31.29 / 0.9614 / 0.150 / 0.1741
Base+CLFT+RHFT+Encoder+Δ	✓	✓	✓	✓	✓	31.32 / 0.9620 / 0.149 / 0.1738
Base+CLFT+RHFT+Encoder+Δ	✓	✓	✓	✓		30.45 / 0.9500 / 0.151 / 0.1743

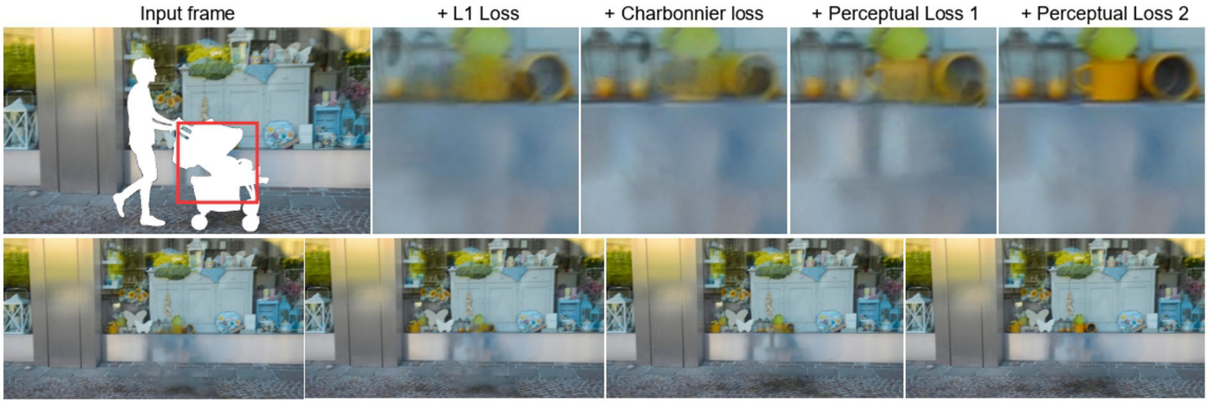

FIGURE 6
Effects of inpainting under different loss functions.

TABLE 3 Scores of STTTN under different loss function combinations.

Method	L1 loss	Charbonnier loss	Perceptual loss 1	Perceptual loss 2	PSNR↑/SSIM↑
L1 loss	✓				29.52 / 0.9395
C loss		✓			30.41 / 0.9532
C loss+P loss 1		✓	✓		31.12 / 0.9598
C loss+P loss 1+P loss 2		✓	✓	✓	31.29 / 0.9614

effect of video repair gradually improves, which proves the effectiveness of each part of the loss function.

5. Conclusion

In this paper, we proposed a novel joint spatial-temporal texture transformer network for video inpainting. Each component cooperates closely, and the repair process progresses layer by layer, making full use of the texture feature information in time and space. The model has outstanding information extraction and content reconstruction capabilities in details and contours, which are essential and suitable for video repair tasks. The excellent results of STTTN's experiments on multiple datasets in multiple scenarios fully demonstrate its superiority over other methods.

However, in our exploration process, we found that STTTN has certain defects and room for further improvement. First, the first defect is also the region normalization defect, that is, color casting is prone to occur. The second defect, which is also a defect that the entire implicit inpainting architecture is prone to have, is the inconsistency in the temporal domain (the before and after frames have abnormal jitter effects during playback due to large local pixel changes, and blur artifacts appear in the video). Video tests appear periodically. We hope that our exploration of video inpainting can help other researchers explore further in this field and that researchers can propose more advanced models to improve the defects we have found thus far.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

Ethics statement

Written informed consent was obtained from the individual(s) for the publication of any potentially identifiable images or data included in this article.

Author contributions

PM provided the idea of the algorithm and designed the entire architecture and was responsible for the writing of the manuscript and the conduct of the experiments. TX reviewed the manuscript. Both authors read and approved the final manuscript.

Funding

This research was supported by the Shaanxi Provincial Technical Innovation Guidance Special (Fund) Plan in 2020 (2020CGXNG-012).

Acknowledgments

I am very grateful to my tutor, TX, for his guidance and help.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Arnab, A., Dehghani, M., Heigold, G., Sun, C., Luči,ć, M., and Schmid, C. (2021). "Vivit: a video vision transformer," in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (Montreal, QC: IEEE), 6836–6846.

Ballester, C., Bertalmio, M., Caselles, V., Sapiro, G., and Verdera, J. (2001). Filling-in by joint interpolation of vector fields and gray levels. *IEEE Trans. Image Process*. 10, 1200–1211. doi: 10.1109/83.935036

Bishop, C. M., Blake, A., and Marthi, B. (2003). "Super-resolution enhancement of video," in *International Workshop on Artificial Intelligence and Statistics* (Palermo), 25–32.

Boßmann, F., Sauer, T., and Sissouno, N. (2019). Modeling variational inpainting methods with splines. *Front. Appl. Math. Stat*. 5, 27. doi: 10.3389/fams.2019.00027

Caelles, S., Montes, A., Maninis, K.-K., Chen, Y., Van Gool, L., Perazzi, F., et al. (2018). The 2018 davis challenge on video object segmentation. *arXiv preprint arXiv:1803.00557*. doi: 10.1109/CVPR.2017.565

Chang, Y.-L., Liu, Z. Y., Lee, K.-Y., and Hsu, W. (2019a). "Free-form video inpainting with 3d gated convolution and temporal patchgan," in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (Seoul), 9066–9075.

Chang, Y.-L., Liu, Z. Y., Lee, K.-Y., and Hsu, W. (2019b). Learnable gated temporal shift module for deep video inpainting. *arXiv preprint arXiv:1907.01131*.

Chang, Y.-L., Liu, Z. Y., Lee, K.-Y., and Hsu, W. (2019c). Learnable gated temporal shift module for deep video inpainting. *arXiv preprint arXiv:1907.01131*. doi: 10.48550/arXiv.1907.01131

Chang, Y.-L., Yu Liu, Z., and Hsu, W. (2019d). "Vornet: spatio-temporally consistent video inpainting for object removal," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*.

Chen, Z., Xie, L., Niu, J., Liu, X., Wei, L., and Tian, Q. (2021). "Visformer: The vision-friendly transformer," in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (Montreal, QC), 589–598.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). "Imagenet: a large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition* (Miami, FL: IEEE), 248–255.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., et al. (2020). An image is worth 16x16 words: transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*. doi: 10.48550/arXiv.2010.11929

Gao, C., Saraf, A., Huang, J.-B., and Kopf, J. (2020). "Flow-edge guided video completion," in *European Conference on Computer Vision* (Glasgow: Springer), 713–729.

Gu, H., Wang, H., Qin, P., and Wang, J. (2022). Chest l-transformer: local features with position attention for weakly supervised chest radiograph segmentation and classification. *Front. Med*. 9, 923456. doi: 10.3389/fmed.2022.923456

Han, K., Wang, Y., Chen, H., Chen, X., Guo, J., Liu, Z., et al. (2020). A survey on visual transformer. *arXiv e-prints, arXiv: 2012.12556*. doi: 10.48550/arXiv.2012.12556

Iizuka, S., Simo-Serra, E., and Ishikawa, H. (2017). Globally and locally consistent image completion. *ACM Trans. Graphics* 36, 1–14. doi: 10.1145/3072959.3073659

Johnson, J., Alahi, A., and Fei-Fei, L. (2016). "Perceptual losses for real-time style transfer and super-resolution," in *European Conference on Computer Vision* (Amsterdam), 694–711.

Kim, D., Woo, S., Lee, J.-Y., and Kweon, I. S. (2019). "Deep video inpainting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Long Beach, CA), 5792–5801.

Kim, T. H., Sajjadi, M. S., Hirsch, M., and Scholkopf, B. (2018). "Spatio-temporal transformer network for video restoration," in *Proceedings of the European Conference on Computer Vision (ECCV)* (Munich), 106–122.

Kobla, V., Doermann, D., Lin, K.-I., and Faloutsos, C. (1996). *Feature normalization for video indexing and retrieval*. Report, Maryland Univ College Park Language and Media Processing Lab.

Lai, W.-S., Huang, J.-B., Ahuja, N., and Yang, M.-H. (2018a). Fast and accurate image super-resolution with deep laplacian pyramid networks. *IEEE Trans. Pattern Anal. Mach. Intell*. 41, 2599–2613. doi: 10.1109/TPAMI.2018.2865304

Lai, W.-S., Huang, J.-B., Wang, O., Shechtman, E., Yumer, E., and Yang, M.-H. (2018b). "Learning blind video temporal consistency," in *Proceedings of the European Conference on Computer Vision (ECCV)* (Munich), 170–185.

Lao, D., Zhu, P., Wonka, P., and Sundaramoorthi, G. (2021). "Flow-guided video inpainting with scene templates," in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (Montreal, QC: IEEE), 14599–14608.

Lee, K., Chang, H., Jiang, L., Zhang, H., Tu, Z., and Liu, C. (2021). Vitgan: Training gans with vision transformers. *arXiv preprint arXiv*:2107.04589.

Lee, S., Oh, S. W., Won, D., and Kim, S. J. (2019). "Copy-and-paste networks for deep video inpainting," in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (Seoul: IEEE), 4413–4421.

Levin, A., Zomet, A., and Weiss, Y. (2003). "Learning how to inpaint from global image statistics," in *Proceedings Ninth IEEE International Conference on Computer Vision, Vol. 1* (Nice: IEEE), 305–312.

Lin, K., Jie, B., Dong, P., Ding, X., Bian, W., and Liu, M. (2022). Convolutional recurrent neural network for dynamic functional mri analysis and brain disease identification. *Front. Neurosci*. 16, 933660. doi: 10.3389/fnins.2022.933660

Liu, R., Deng, H., Huang, Y., Shi, X., Lu, L., Sun, W., et al. (2021a). Decoupled spatial-temporal transformer for video inpainting. *arXiv preprint arXiv:2104.06637*. doi: 10.48550/arXiv.2104.06637

Liu, R., Deng, H., Huang, Y., Shi, X., Lu, L., Sun, W., et al. (2021b). "Fuseformer: fusing fine-grained information in transformers for video inpainting," in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (Montreal), 14040–14049.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., et al. (2021c). "Swin transformer: hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (Montreal, QC: IEEE), 10012–10022.

Oh, S. W., Lee, S., Lee, J.-Y., and Kim, S. J. (2019). "Onion-peel networks for deep video completion," in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (Seoul), 4403–4412.

Perazzi, F., Pont-Tuset, J., McWilliams, B., Van Gool, L., Gross, M., and Sorkine-Hornung, A. (2016). "A benchmark dataset and evaluation methodology for video object segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Las Vegas, NV: IEEE), 724–732.

Shu, X., Zhang, L., Qi, G.-J., Liu, W., and Tang, J. (2021). Spatiotemporal co-attention recurrent neural networks for human-skeleton motion prediction. *IEEE Trans. Pattern Anal. Mach. Intell*. 44, 3300–3315. doi: 10.1109/TPAMI.2021.3050918

Suin, M., Purohit, K., and Rajagopalan, A. (2021). "Distillation-guided image inpainting," in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (Montreal, QC: IEEE), 2481–2490.

Tang, J., Shu, X., Yan, R., and Zhang, L. (2019). Coherence constrained graph lstm for group activity recognition. *IEEE Trans. Pattern Anal. Mach. Intell*. 44, 636–647. doi: 10.1109/TPAMI.2019.2928540

Telea, A. (2004). An image inpainting technique based on the fast marching method. *J. Graph. Tools* 9, 23–34. doi: 10.1080/10867651.2004.10487596

Wang, C., Huang, H., Han, X., and Wang, J. (2019). Video inpainting by jointly learning temporal structure and spatial details. *Proc. AAAI Conf. Artif. Intell*. 33, 5232–5239. doi: 10.1609/aaai.v33i01.33015232

Wang, C., and Wang, Z. (2022). Progressive multi-scale vision transformer for facial action unit detection. *Front. Neurorob*. 15, 824592. doi: 10.3389/fnbot.2021.824592

Wang, N., Ma, S., Li, J., Zhang, Y., and Zhang, L. (2020). Multistage attention network for image inpainting. *Pattern Recogn*. 106, 107448. doi: 10.1016/j.patcog.2020.107448

Wang, T.-C., Liu, M.-Y., Zhu, J.-Y., Liu, G., Tao, A., Kautz, J., et al. (2018). Video-to-video synthesis. *arXiv preprint arXiv:1808.06601*. doi: 10.48550/arXiv.1808.06601

Xu, N., Yang, L., Fan, Y., Yue, D., Liang, Y., Yang, J., et al. (2018). Youtube-vos: a large-scale video object segmentation benchmark. *arXiv preprint arXiv*:1809.03327. doi: 10.1007/978-3-030-01228-1_36

Xu, R., Li, X., Zhou, B., and Loy, C. C. (2019). "Deep flow-guided video inpainting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Seoul), 3723–3732.

Yang, F., Yang, H., Fu, J., Lu, H., and Guo, B. (2020). "Learning texture transformer network for image super-resolution," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Seattle, WA), 5791–5800.

Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., and Huang, T. S. (2019). "Free-form image inpainting with gated convolution," in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (Seoul: IEEE), 4471–4480.

Yu, T., Guo, Z., Jin, X., Wu, S., Chen, Z., Li, W., et al. (2020). Region normalization for image inpainting. *Proc. AAAI Conf. Artif. Intell.* 34, 12733–12740. doi: 10.1609/aaai.v34i07.6967

Zeng, Y., Fu, J., and Chao, H. (2020). "Learning joint spatial-temporal transformations for video inpainting," in *European Conference on Computer Vision* (Glasgow), 528–543.

Zou, X., Yang, L., Liu, D., and Lee, Y. J. (2021). "Progressive temporal feature alignment network for video inpainting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16448–16457.