Video Inpainting by Jointly Learning Temporal Structure and Spatial Details

Chuan Wang[1]   **Haibin Huang**[1]   **Xiaoguang Han**[‡2]   **Jue Wang**[1]

Megvii (Face++)[1]
{wangchuan, huanghaibin, wangjue}@megvii.com
Shenzhen Research Inst. of Big Data, The Chinese University of Hong Kong, Shenzhen, China[2]
hanxiaoguang@cuhk.edu.cn

Abstract

We present a new data-driven video inpainting method for recovering missing regions of video frames. A novel deep learning architecture is proposed which contains two sub-networks: a temporal structure inference network and a spatial detail recovering network. The temporal structure inference network is built upon a 3D fully convolutional architecture: it only learns to complete a low-resolution video volume given the expensive computational cost of 3D convolution. The low resolution result provides temporal guidance to the spatial detail recovering network, which performs image-based inpainting with a 2D fully convolutional network to produce recovered video frames in their original resolution. Such two-step network design ensures both the spatial quality of each frame and the temporal coherence across frames. Our method jointly trains both sub-networks in an end-to-end manner. We provide qualitative and quantitative evaluation on three datasets, demonstrating that our method outperforms previous learning-based video inpainting methods.

Introduction

Given an image or a video with holes inside (e.g. generated by object removal), inpainting (also called completion) techniques try to recover the missing video content to produce a natural looking result. This problem has drawn great attention in the past two decades due to strong industrial demands on image and video editing applications. Inpainting is a very challenging task as the requirements are two-fold: (1) the generated content in the missing regions must be semantically correct given their surrounding content; and (2) the missing regions need to be filled in a seamless way so that the original holes are visually unnoticeable.

In this work, we focus on video inpainting, an extended problem from image inpainting with the added temporal dimension. Such extension brings new technical challenges that are difficult to resolve. First, recovering missing video content requires the understanding of not only the spatial context of each frame, but also the motion context across frames. Second, the output video must maintain high spatio-temporal consistency, in both global context-level and local image-feature-level. Although there
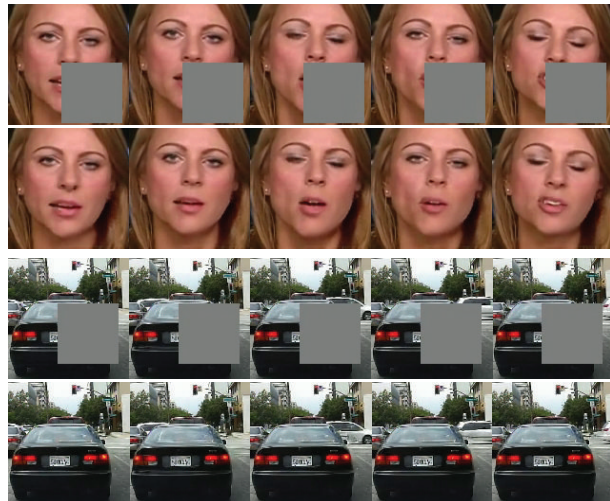

Figure 1: Video inpainting results by our approach. Row 1, 3: input frames with missing regions (shown in gray). Row 2, 4: our results. Note that the filled regions contain rich image details and are temporally coherent.

has been tremendous progress in image inpainting, e.g. patch-based image synthesis (Barnes et al. 2009; 2010) and deep learning based approaches (Pathak et al. 2016; Iizuka, Simo-Serra, and Ishikawa 2017), a direct extension of those methods in 3D would not work well for video inpainting. Specifically, (Wexler, Shechtman, and Irani 2007; Venkatesh, Cheung, and Zhao 2009; Newson et al. 2014; Huang et al. 2016) have tried to extend spatial 2D patch synthesis to spatial-temporal 3D patch synthesis. However, local synthesis, even in 3D space, cannot guarantee global semantic correctness. Moreover, directly applying image inpainting networks to each frame individually often leads to temporally jittering results that are visually unacceptable, as we will demonstrate later.

We present a novel end-to-end deep learning architecture to tackle the above issues for high quality video inpainting. As shown in Fig. 2, the network consists of a temporal structure prediction sub-network and a spatial detail recovering sub-network. The former sub-network treats a video as a 3D volume. It takes a down-sampled version of the origi-

‡Corresponding Author.

nal video as input, and fill the holes in it using 3D CNN with an Encoder-Decoder architecture. We use this output volume as temporal structure guidance, since it captures the motion structure across time but lacks spatial details. The spatial detail inference network then takes the original video and the temporal structure guidance as input, and generates completed video frames in their original resolution. It has a 2D Encoder-Decoder architecture with global and local l_1 consistency losses. These two sub-networks are jointly trained and can benefit from each other. In other words, the temporal structure guidance improves both the temporal smoothness and the context consistency of the final video. Meanwhile, the loss of the spatial detail recovering network is also back-propagated into the first network and helps improve the accuracy of temporal structure prediction.

In summary, our main contributions are:

- it is the first work to use deep neural networks for solving the problem of video completion. Compared with existing methods, the proposed algorithm can deal with the video with complex appearances and large missing regions;
- we design a novel deep learning architecture that uses 3D CNN for temporal structure prediction and 2D CNN for spatial detail recovering, where the output temporal structure is fused into the 2D CNN to guide the detail inference;
- we perform joint training of the two sub-networks, which further improves the performance of the overall system.

Related Work

In this section, we introduce the related works in the following three aspects and refer the readers to the works of (Chhabra and Birchha 2014) and (Ilan and Shamir 2015) for detailed literature review of image/video inpainting.

Patch-based image/video inpainting. To fill in the holes using patch-based synthesis is the most used traditional strategy for image inpainting. This was firstly proposed in (Efros and Leung 1999) and (Efros and Freeman 2001), where the missing contents are recovered in a region-growing way: the method starts from boundary of holes and extends the region by searching appropriate patches and assembling them together. Following up this work, there are many different directions for improvement in searching and optimization (Kwatra et al. 2005; Wexler, Shechtman, and Irani 2007; Simakov et al. 2008; Barnes et al. 2009; 2010; 2015). It is also adapted to video inpainting problem by replacing 2D patch synthesis with 3D spatial-temporal patch synthesis across frames. This was firstly proposed in (Wexler, Shechtman, and Irani 2004; 2007) to ensure the temporal consistency of the generated video and later improved in (Jia, Hu, and Martin 2005; Patwardhan, Sapiro, and Bertalmio 2005; Venkatesh, Cheung, and Zhao 2009) to handle more complicated video input. However, all of these works are designed for the video with repeated content across frames. They are unable to tackle the problem we proposed in this paper where missing parts cannot be replaced by similar content in the input. Resorting to a large video dataset, we

try to train a CNN in this work for missing contents prediction based on the high-level context understanding.

Image completion using 2D CNN. Recently, Convolutional Neural Network was firstly used in (Xie, Xu, and Chen 2012) for image inpainting but only for small holes. Pathak et al. (Pathak et al. 2016) then proposed to deal with large missing regions using an encoder-decoder architecture which can efficiently learn the context feature of the image. For high-resolution image inpainting, Yang et al. (Yang et al. 2017) developed a multi-scale neural patch synthesis algorithm that not only preserves contextual structure but also produces high-frequency details. The algorithm proposed in (Iizuka, Simo-Serra, and Ishikawa 2017) further improves the performance by involving two adversarial losses to measure both the global and local consistency of the result. Different from the previous works which only focus on box-shaped holes, it also develops a strategy to handle the holes with arbitrary shapes. To extend these methods from image to video domain is a challenging task, because video completion not only needs to have an accurate context understanding of both frames and motions, but also requires to ensure temporal smoothness of the output. In this paper, we propose a novel deep learning architecture for this problem which takes use of both 2D and 3D CNNs to jointly learn the temporal structure and spatial details.

Shape completion using 3D CNN. Another series of works related to our paper is using 3D CNN for 3D shape completion. Similar to deep learning based image inpainting, most of the methods such as (Sharma, Grau, and Fritz 2016; Varley et al. 2017; Dai, Qi, and Nießner 2017) uses encoder-decoder architecture but with 3D CNN for solving this problem. However, all these techniques can only handle low-resolution grids (typically 30^3 voxels) due to the high computational cost of 3D convolutions. To address this issue, many approaches are proposed most recently. Resorting to a dataset, Dai et al. (Dai, Qi, and Nießner 2017) used patch retrieval and assembly as a post-processing to refine the low-resolution output of encoder-decoder network. For such post-refinement, the method in (Wang et al. 2017b) proposed a strategy to slice the low-resolution output into a sequence of images and did super-resolution and completion for each sliced image with a recurrent neural network. Han et al. (Han et al. 2017) designed a hybrid networks for jointly global structure prediction and local geometry inference. Our work is inspired by this method but differs from it in two aspects. Firstly, the method of (Han et al. 2017) conducted the completion in a region-growing way while ours is an end-to-end architecture for completion. Secondly, for details inference, the method in (Han et al. 2017) only looks at a local region which lacks much surrounding context information while our algorithm uses the content of the whole image for missing information recovering.

Algorithm

Our method is built upon deep neural networks. It takes an incomplete video V_{in} and a mask video M as input, and

Note: The gray blocks representing "Input Video" and "Input Frame" contains 4 channels, R/G/B and mask.
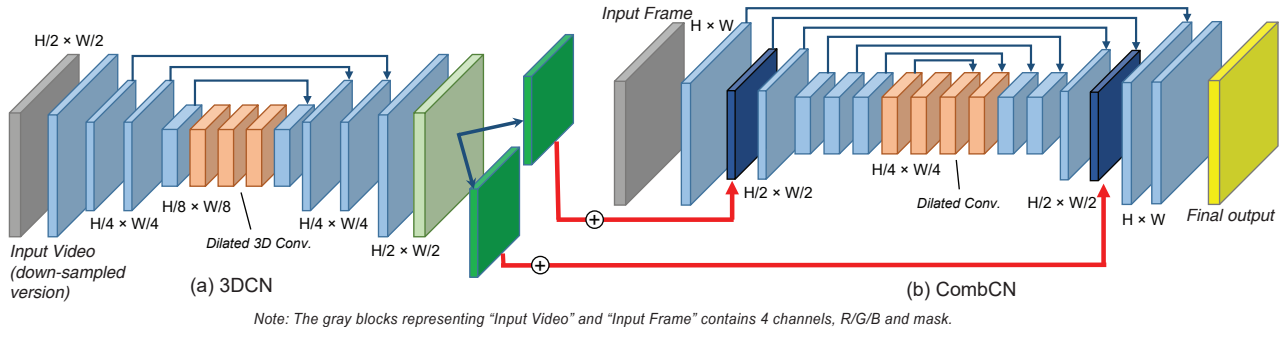
Figure 2: Network architecture of our 3D completion network (3DCN) and 3D-2D combined completion network (CombCN). The 3DCN works in low resolution, producing an inpainted video as output. Its individual frames are further convolved and added into the first and last layer of the same size in CombCN. The input video for 3DCN and the input frame for CombCN, shown as gray blocks, are in 4-channel format, containing RGB and the mask indicating the holes to be filled.

produces an complete video V_{out} as output. The incomplete video is represented as a $F \times H \times W$ volume, where F, H, W are number of frames, height and width of V_{in}. M and V_{out} are of the same size as V_{in}. In training stage, V_{in} is obtained by randomly generating holes on a complete video V_c, i.e. the target ground truth of the inpainting task.

To produce a spatial-temporal coherent inpainted video, our network consists of two sub-networks. One network is a 3D completion network (3DCN), which utilizes 3D CNN to infer the temporal structure globally from a down-sampled version ($F \times \frac{H}{r} \times \frac{W}{r}$) of V_{in} and M. In this paper, we use superscript d to distinguish the downsampled videos from their original versions. So 3DCN takes V_{in}^d and M^d as input and produces V_{out}^d as output. Another sub-network is a 3D-2D combined completion network (CombCN). It applies 2D CNN to perform completion frame by frame. The input of this network includes one incomplete but high-resolution frame of V_{in} with its mask frame of M, and a low-resolution but complete frame $I_{out}^{d,k}$ of V_{out}^d ($k = 1, 2, ..., F$). In our paper, F, H, W, r are set to 32, 128, 128 and 2 respectively.

Temporal structure inference by 3DCN

A video inpainting task requires not only filling the missing part within each frame but also keeping the consistency between successive frames. For this reason, it would fail if directly apply the existing image inpainting methods such as (Iizuka, Simo-Serra, and Ishikawa 2017) to video frames separately, because they lack the mechanism of preserving temporal coherence. On the other hand, video can also be viewed as a spatial-temporal volume and its temporal structure could be preserved when algorithms are globally applied to it, such as (Wang et al. 2017a; Hara, Kataoka, and Satoh 2017; Wang et al. 2014). To this end, we apply 3D CNN globally to video inpainting. However, due to the expensive memory cost of 3D convolutions, we only utilize 3D CNN on a down-sampled version V_{in}^d of the input video. The 3D completion network can generate an inpainted video V_{out}^d which captures the temporal structure of the original video, even though its individual frame lacks details.

Our 3D completion network follows an encoder-decoder structure and consists of 12 layers totally. Given the incomplete video and its mask as input, it first exploits 4 strided convolutional layers to encode it to a latent space, capturing its temporal-spatial structure. Then 3 dilated convolutional layers with rate 2, 4, 8 are followed to capture the spatial-temporal information in larger perception field. At last, video inpainting is finally achieved by 3 convolutional and 2 fractionally-strided convolutional layers in an alternative order, yielding the result with missing part filled. Rather than using the max-pooling and upsampling layers to compute the feature maps, we employ 3×3 convolution kernels with stride of 2, which ensures that every pixel contributes. Meanwhile, considering that non-successive frames may have loose relations with the current one, and to avoid information loss across frames, we limit stride and dilation to take effects only within frame rather than across frames. As a result, the feature map of each layer has a constant frame number F. Besides, all the convolutional layers are followed by batch normalization (BN) and ReLU non-linearity activation except the last one. Paddings are involved to make the input and output have exactly the same size. The skip-connections as U-Net (Ronneberger, Fischer, and Brox 2015) are also employed to facilitate the feature mixture across encoder and decoder. The detailed configuration of our 3D completion network is illustrated in Fig. 2 (a) and listed in Table 1 (top) (BN and ReLU are not shown for brevity).

Training. Let $G_v(V_{in}^d, M^d) = V_{out}^d$ denotes the 3DCN in a functional form. The binary masks M^d and M take the value 1 inside regions to be filled-in and 0 elsewhere. The pixels of V_{in}^d and V_{in} inside the mask region are pre-filled with the mean pixel value of the training dataset before feeding it to the network. During training, we minimize the l_1 norm the difference between V_{out}^d and V_c^d. The difference is also weighted considering the completion region mask is used. Specifically, the l_1 loss of 3DCN is defined by:

$$L^{3DCN}(V_{in}^d, M^d, V_c^d) = \frac{\|M^d \odot (G_v(V_{in}^d, M^d) - V_c^d)\|}{\|M^d\|}$$

(1)

Layer No.	Type	Kernel	Stride	Channel	Dilation	Layer No.	Type	Kernel	Stride	Channel	Dilation
1	conv.	5	1	16	-	7	dilated conv.	3	1	256	8
2	conv. ↓	3	2	32	-	8	conv.	3	1	128	-
3	conv.	3	1	64	-	9	deconv. ↑	4	2	64	-
4	conv. ↓	3	2	128	-	10	conv.	3	1	32	-
5	dilated conv.	3	1	256	2	11	deconv. ↑	4	2	16	-
6	dilated conv.	3	1	256	4	12	conv.	3	1	3	-

Layer No.	Type	Kernel	Stride	Channel	Dilation	Layer No.	Type	Kernel	Stride	Channel	Dilation
1	conv.	5	1	64	-	10	dilated conv.	3	1	256	16
2*	conv. ↓	3	2	128	-	11	conv.	3	1	256	-
3	conv.	3	1	128	-	12	conv.	3	1	256	-
4	conv. ↓	3	2	256	-	13	deconv. ↑	4	2	128	-
5	conv.	3	1	256	-	14	conv.	3	1	128	-
6	conv.	3	1	256	-	15*	deconv. ↑	4	2	64	-
7	dilated conv.	3	1	256	2	16	conv.	3	1	32	-
8	dilated conv.	3	1	256	4	17	conv.	3	1	3	-
9	dilated conv.	3	1	256	8						

Table 1: Network architecture of 3DCN (top) and CombCN (bottom). In the bottom table, * represents the layers where combination takes place in CombCN.

where $\odot$ is the pixelwise multiplication and $\|\cdot\|$ is the l_1 norm.

Spatial details inference by CombCN

The output of 3DCN is a low resolution inpainted video. It conveys temporal structure but lacks details within each frame. To restore the details, we extend a 2D completion network (2DCN) inspired by a state-of-the-art image inpainting work (Iizuka, Simo-Serra, and Ishikawa 2017), obtaining a combined completion network (CombCN). This CombCN consists of 17 layers including 11 strided convolutional layers, 2 fractional deconvolutional layers and 4 dilated convolutional layers. It also follows an encoder-decoder structure where the minimal feature map size is $\frac{H}{4} \times \frac{W}{4}$. The dilated convolutional layers are involved to obtain a larger perception field so that the network can "see" areas far from the missing part. The configuration of CombCN is listed in Table. 1 (bottom) and we encourage the readers to review (Iizuka, Simo-Serra, and Ishikawa 2017) for a detailed explanation of its original configuration. Note that we also made a modification by involving skip-connection as U-Net.

To tackle the issue that 2DCN treats each frame independently without considering temporal coherence, we also inject the information from the output of 3DCN to CombCN. This is achieved by using two convolutional layers to extract two feature maps of the 3DCN output separately. The two feature maps are then added to the first and last layer of the same size in CombCN, serving as a temporal guidance. In this paper, since 3DCN works on videos of size $\frac{H}{2} \times \frac{W}{2}$, the combination takes place in the 2nd and the 15th layer. We compared the effectiveness of this combination setting with the basic 2DCN on successive frames. The experimental results illustrate temporal coherence can be well preserved when inpainting frames separately, as shown in Figs. 3 and 1.

Training. Let $G_i(V_{in}^k, M^k, I_{out}^{d,k}) = I_{out}^k$ denotes the CombCN in a functional form, where V_{in}^k, M^k and $I_{out}^{d,k}$ are the k-th frame of the incomplete video V_{in}, mask video M and the inpainted video V_{out}^d by 3DCN. During training, we

view a video data sample as a batch of images so that the data format can be well supported by the existing deep learning frameworks like TensorFlow. For a video data sample, the optimization goal is to minimize the mean of the l_1 norm of the difference between I_{out}^k and V_c^k ($k = 1, 2, ..., F$). Specifically, the loss of CombCN is defined as:

$$L^{CombCN}(V_{in}, M, V_{out}^d, V_c) \qquad (2)$$
$$= \sum_{k=1,2,...,F} \frac{\|M^k \odot (G_i(V_{in}^k, M^k, I_{out}^{d,k}) - V_c^k)\|}{F \cdot \|M^k\|}$$

In practice, we first pre-train 3DCN to converge, and then train CombCN with the pre-trained 3DCN model finetuned. This training strategy can lead to fast convergence of CombCN compared with training both sub-networks together from scratch. We also enable finetuning 3DCN in order to acquire lower loss compared with the strategy without finetuning. In this case, we jointly optimize the weighted sum of the two sub-network losses, i.e.

$$L^{total} = L^{3DCN} + \alpha L^{CombCN} \qquad (3)$$

where α is a balancing parameter which is set to 1.0 in our paper. A detailed comparison of different training strategies are presented in Section *Performance of variants of training strategy*. The experiment result shows that our training strategy outperforms the other two, i.e. pre-training 3DCN disabled and finetuning 3DCN disabled when training CombCN.

Experimental Results

Dataset and implementation details

To validate our 3D-2D combined completion network, we tested on three datasets, FaceForensics (Rössler et al. 2018), 300VW (Chrysos et al. 2015) and Caltech (Dollár et al. 2012).

The first two datasets contain 1,004 and 300 video clips with human faces respectively, where the faces are near-frontal pose and neutral expression change across frames. To further stress test our method, we also run on Caltech (Dollár et al. 2012) which contains 10 hours of video of size
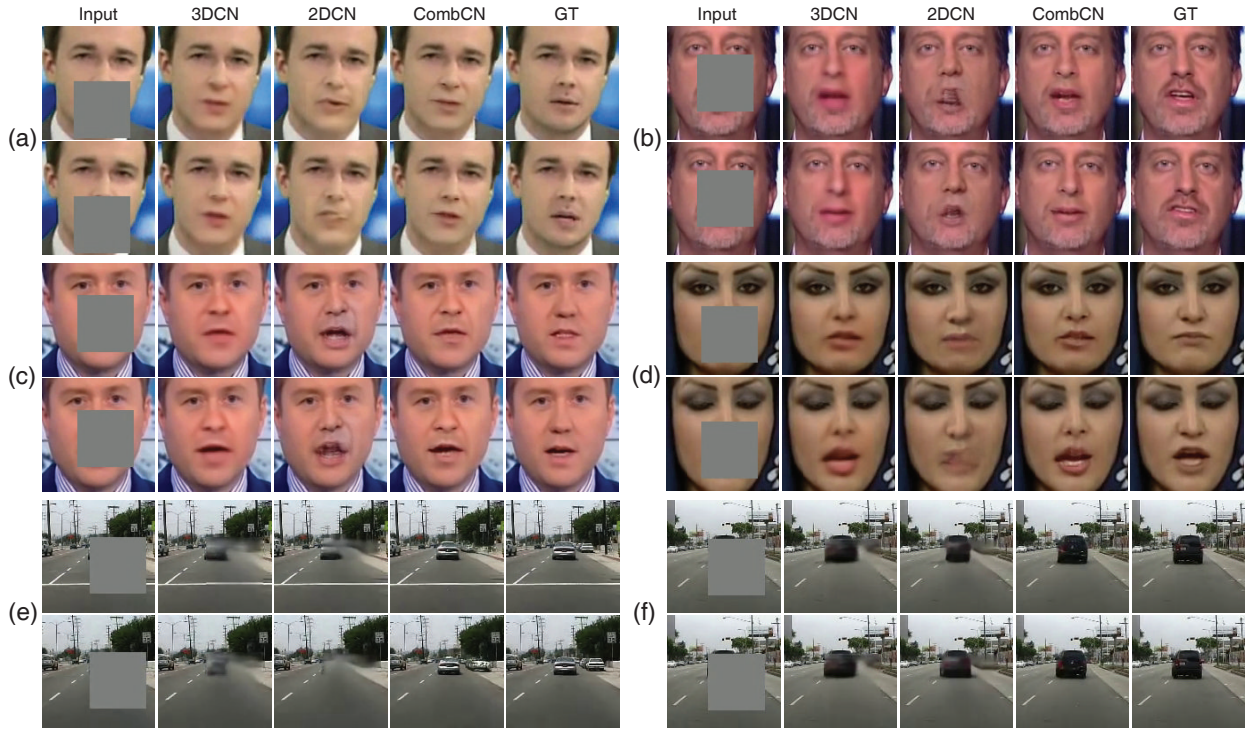
Figure 3: Inpainted frames on datasets FaceForensics (a∼d) and Caltech (e, f). In each panel, the two rows represent two frames of a video, and the five columns from left to right are input, results by 3DCN, 2DCN and CombCN, as well as the target ground truth. Better visual experience can be obtained in our accompanying supplemental materials.

640×480 taken from a vehicle driving through regular traffic in an urban environment. Compared with the face videos with obvious semantic structures, Caltech dataset is more challenging to an inpainting task.

In data preparation stage, we group every 32 frames from the original video clips into a data sample. Each frame is in 128^2 resolution, which is generated in the following manner. For the face videos, i.e. FaceForensics and 300VW, we first crop the face out with a squared bounding box; while for Caltech dataset, we directly crop the central 480^2 region out. The cropped region is then resized into 128^2 resolution. For each dataset, we separate the whole data samples into training and validation sets and control their proportion $5 : 1$.

During training we randomly generate a hole across all frames in the $[0.375l, 0.5l]$ pixel range and fill it with the mean pixel value of the training dataset, where l is the frame size (128 in this paper). The range follows the same ratio as in (Iizuka, Simo-Serra, and Ishikawa 2017). The position of the hole for a video data sample is identical for all of its frames. Based on these inputs, we first pre-train 3DCN to convergence and then train CombCN with the pre-trained 3DCN model jointly finetuned. The CombCN is trained with $100k$ iterations by an Adam optimizer, whose regression weight and learning rate are set to 0.01 and 0.001, respectively. Each iteration costs approximate 0.8s and it takes nearly 30 hours to complete the entire training. The detailed configuration of 3DCN and CombCN is illustrated in Fig. 2. The implementation is based on TensorFlow and the net-

work training is performed on a single NVIDIA GeForce GTX 1080 Ti.

Comparisons with existing methods

Our results in comparison with those produced by 2DCN and 3DCN can be found in Figs. 3 and 4. More video results are also presented in our accompanying supplemental materials.

Ours vs. 2DCN. We first compare our method with a state-of-the-art image inpainting method (Iizuka, Simo-Serra, and Ishikawa 2017), which is applied to each video frame independently. Such an image inpainting method is expected to produce higher quality result, but when applying it to each video frame, we found that the quality of frames are unstable and annoying visual artifacts like flicker are present. This is because it lacks the mechanism to preserve the spatial-temporal coherence of the video. For example, in the 3rd column of Fig. 3(a)(d), the algorithm produces high quality results on the first frame but fails in the second frame, making the mouths missing. This circumstance also occurs in the Caltech dataset as shown in Fig. 3(e), where the car disappears unreasonably in the second frame. In addition, if there is motion in video, 2DCN may also produce blurry or distorted results as illustrated in Fig. 3(b)(c). In contract, our CombCN produces reasonable and stable results (the 4th column) across frames. This temporal coherence ensures the pleasant user experience when the video being played.
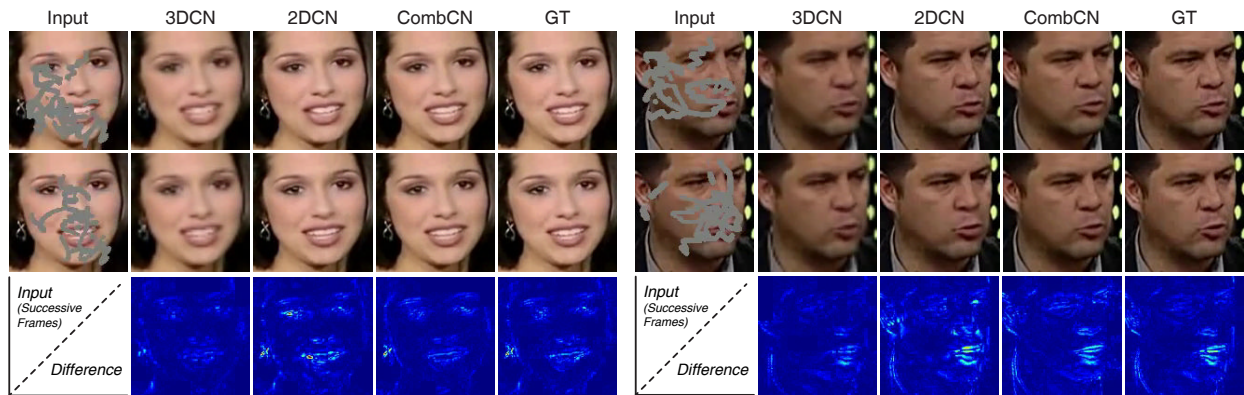
Figure 4: Inpainting results from videos with random holes. We visualize the differences between the two successive frames to illustrate the inter-frame consistency. It shows that our CombCN produced clearer results than 3DCN, and smoother results than 2DCN. Better visual experience can be obtained in our accompanying supplemental materials.

	3DCN	2DCN	CombCN (ours)		
FaceForensics	7.18	6.77	**6.27**		
Caltech	11.91	11.16	**9.56**		

	V-1	V-2	T-1	T-2	our method
3DCN	9.51	11.56	6.30	9.28	**4.45**
CombCN	6.39	8.13	5.18	6.31	**4.20**

Table 2: Final l_1 losses. Top: the losses of 3DCN, 2DCN and CombCN of datasets FaceForensics and Caltech. Bottom: the losses of 3DCN and CombCN in 300VW dataset, based on variants of 3DCN (V-1, V-2) and training strategy (T-1, T-2), in comparison with our method.

Ours vs. 3DCN. We also compare our results with the low resolution output achieved by 3DCN. They are listed in the 2nd column in Fig. 3. Due to the usage of 3D CNN in addition to working on low resolution, the inpainted frames contain significant blurry artifacts so that details cannot be well reconstructed. For example in Fig. 3(b)(c), the teeth are almost missing in the inpainted frames by 3DCN. However, unlike the results by 2DCN, they have smooth transition across frames. In comparison, our approach preserves temporal coherence and details simultaneously. The final l_1 losses by 3DCN, 2DCN and CombCN are listed in Table. 2 (top).

Random holes. Our system can also be easily applied to the inpainting task with random holes in validation/testing phase, even in the case that holes are distinct for the input frames. Note that in this case, though working in a lower resolution, 3DCN can further reveal its power to fill the holes in a temporally consistent manner, because it can take advantage of the pixels of the non-hole regions in the contiguous frames. However, distinct holes make it more challenging for 2DCN to keep the inter-frame consistency. As a result, our CombCN combines the two benefits from the two subnetworks and is able to produce clearer and smoother results.

The value of l_1 loss in this paper has been normalized so that it is equal to the mean error for each pixel. Its value range is $[0, 255]$.

Fig. 4 shows two examples of video inpainting with random holes.

Ablation studies

To discover the vital elements in the success of our proposed model for video inpainting, we made two groups of variants of our method. They are based on modifications of the 3D-CN structure and the training strategy separately. The final losses of all variants and our method are listed in Table. 2 (bottom). These ablation studies were conducted on 300VW dataset.

Performance of variants of 3DCN To investigate the influence of 3DCN to the final results by CombCN, we first modified the structure of 3DCN to its two variants V-1, V-2 as below.

V-1. Feed 3DCN with videos in lower resolution.
In this experiment, we fed the 3DCN with a lower resolution version of the original video, i.e. setting down-sample rate $r = 4$ to produce a 32^3 video $\tilde{V}_{in}^d$. Accordingly, we changed the combination layers in CombCN to the first and last feature maps of size 32^2 instead of 64^2. In this setting, the 3DCN produces more blurry frames while the temporal coherence is rarely lost. Our experiment shows that the convergence errors of 3DCN and CombCN are 9.51 and 6.39, while our baseline model produces the corresponding errors 4.45 and 4.20.

V-2. Involve down-sampling in time axis in 3DCN.
We alternatively modified our basic 3DCN to allow strided convolutions across time-axis. So the frame number of the feature maps in the 2nd and 4th convolutional layers are down-sampled, becoming $\frac{F}{2}$ and $\frac{F}{4}$ respectively. The deconvolutional layers are also modified to support up-sampling in time-axis. In this setting, the temporal coherence of the inpainted frames by 3DCN are less preserved while the extracted features become more compact. Our experiment shows that visually this setting does not obviously down-grade the performance of CombCN, and the
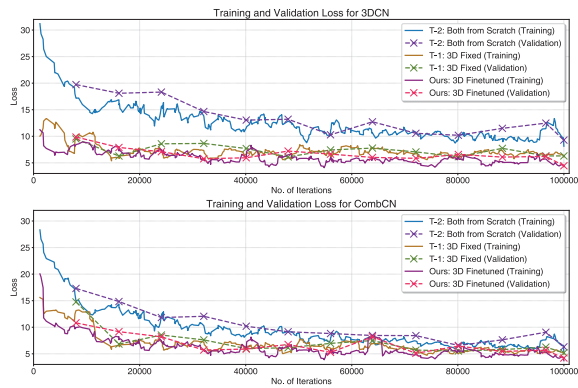
Figure 5: Training and validation loss of variants of training strategy for 3DCN (top) and CombCN (bottom).

convergence errors increase to 11.56 for 3DCN and 8.13 for CombCN.

Performance of variants of training strategy As aforementioned, our experimental results were produced by a CombCN trained with a fine-tuned pre-trained model of 3D-CN. To investigate the performance of this training strategy, we further compared two other strategies T-1 and T-2 as follows.

T-1. Pre-train 3DCN, then train CombCN without fine-tuning it.
We first disable fine-tuning the pre-trained model of the 3DCN when training the CombCN. In this setting, the pre-trained 3DCN produces inpainted video frames in low resolution and they are directly fed forward to the CombCN. Due to lack of parameter updating, the loss of 3D-CN keeps fluctuating without dropping, while the loss of the CombCN drops to convergence after nearly $20k$ iterations. After nearly $100k$ iterations, the loss approaches the value that is achieved by the strategy of fine-tuning enabled. We plot the two losses of 3DCN and CombCN in $100k$ iterations labelled by "T-1: 3D Fixed" in Fig. 5.

T-2. Train 3DCN and CombCN jointly from scratch.
We further remove the stage of pre-training 3DCN, and train 3DCN and CombCN jointly from scratch. Compared with our method and T-1, due to the lack of rich informative guidance produced by 3DCN, it takes considerable iterations (over $50k$) for CombCN and 3DCN to converge. Furthermore, the convergence losses of the two sub-networks are also higher, i.e. 9.28 for 3DCN and 6.31 for CombCN. The result reveals the capability of our current training strategy in this paper.

Limitations

Our approach can handle videos with unstructured information and motion in most cases. However, it may still fail if the test video has a severe variation from the training data. Furthermore, inferring temporal coherence relies on 3DCN-N so that large motion cannot be easily captured due to the limitation of the size of receptive field. Fig. 6 illustrates an example of our failure case, where the test video displays a
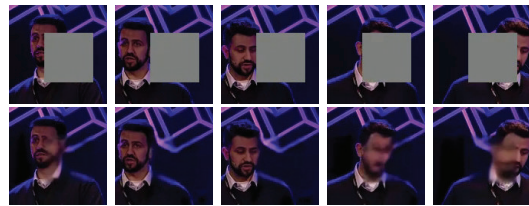

Figure 6: Failure case. Top: five frames with holes as input. Bottom: inpainted results by CombCN.

man with large motion and was captured in a different view setting (far from the face). As a result, our approach produces an unreasonable face in the 4th frame. We believe using an optical flow and LSTM based solution as in (Lai et al. 2018; Ren et al. 2016) could be a potential idea for this problem.

Moreover, unlike state-of-the-art image inpainting approaches commonly involving GAN to synthesize more vivid results, we only use l_1 loss in our paper, which may potentially limit the power of our combination idea. We will leave it as a future work about how to integrate GANs to our 3DCN and CombCN.

Conclusion

We have presented an end-to-end framework for video inpainting through a joint 2D-3D CNN which contains a temporal structure inference network and spatial detail recovering network. Our method can fill regular or random holes across frames to produce plausible results. These results show that our method significantly improves the performance of existing methods. We also believe this architecture has potentials to be applied to other video generation tasks.

Acknowledgements

References

Barnes, C.; Shechtman, E.; Finkelstein, A.; and Goldman, D. B. 2009. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics-TOG* 28(3):24.

Barnes, C.; Shechtman, E.; Goldman, D. B.; and Finkelstein, A. 2010. The generalized patchmatch correspondence algorithm. In *European Conference on Computer Vision*, 29–43. Springer.

Barnes, C.; Zhang, F.-L.; Lou, L.; Wu, X.; and Hu, S.-M. 2015. Patchtable: Efficient patch queries for large datasets and applications. *ACM Transactions on Graphics (TOG)* 34(4):97.

Chhabra, J. K., and Birchha, M. V. 2014. Detailed survey on exemplar based image inpainting techniques. *International Journal of Computer Science and Information Technologies* 5(5):6350–635.

Chrysos, G. G.; Antonakos, E.; Zafeiriou, S.; and Snape, P. 2015. Offline deformable face tracking in arbitrary videos. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 1–9.

Dai, A.; Qi, C. R.; and Nießner, M. 2017. Shape completion using 3d-encoder-predictor cnns and shape synthesis. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 3.

Dollár, P.; Wojek, C.; Schiele, B.; and Perona, P. 2012. Pedestrian detection: An evaluation of the state of the art. *PAMI* 34.

Efros, A. A., and Freeman, W. T. 2001. Image quilting for texture synthesis and transfer. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, 341–346. ACM.

Efros, A. A., and Leung, T. K. 1999. Texture synthesis by non-parametric sampling. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, 1033–1038. IEEE.

Han, X.; Li, Z.; Huang, H.; Kalogerakis, E.; and Yu, Y. 2017. High-resolution shape completion using deep neural networks for global structure and local geometry inference. In *IEEE International Conference on Computer Vision (ICCV)*.

Hara, K.; Kataoka, H.; and Satoh, Y. 2017. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? *arXiv preprint* arXiv:1711.09577.

Huang, J.-B.; Kang, S. B.; Ahuja, N.; and Kopf, J. 2016. Temporally coherent completion of dynamic video. *ACM Transactions on Graphics (TOG)* 35(6):196.

Iizuka, S.; Simo-Serra, E.; and Ishikawa, H. 2017. Globally and Locally Consistent Image Completion. *ACM Transactions on Graphics (Proc. of SIGGRAPH 2017)* 36(4):107.

Ilan, S., and Shamir, A. 2015. A survey on data-driven video completion. In *Computer Graphics Forum*, volume 34, 60–85. Wiley Online Library.

Jia, Y.-T.; Hu, S.-M.; and Martin, R. R. 2005. Video completion using tracking and fragment merging. *The Visual Computer* 21(8-10):601–610.

Kwatra, V.; Essa, I.; Bobick, A.; and Kwatra, N. 2005. Texture optimization for example-based synthesis. In *ACM Transactions on Graphics (ToG)*, volume 24, 795–802. ACM.

Lai, W.-S.; Huang, J.-B.; Wang, O.; Shechtman, E.; Yumer, E.; and Yang, M.-H. 2018. Learning blind video temporal consistency. In *European Conference on Computer Vision*.

Newson, A.; Almansa, A.; Fradet, M.; Gousseau, Y.; and Pérez, P. 2014. Video inpainting of complex scenes. *SIAM Journal on Imaging Sciences* 7(4):1993–2019.

Pathak, D.; Krahenbuhl, P.; Donahue, J.; Darrell, T.; and Efros, A. A. 2016. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2536–2544.

Patwardhan, K. A.; Sapiro, G.; and Bertalmio, M. 2005. Video inpainting of occluding and occluded objects. In *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, volume 2, II–69. IEEE.

Ren, J. S.; Hu, Y.; Tai, Y.-W.; Wang, C.; Xu, L.; Sun, W.; and Yan, Q. 2016. Look, listen and learn - a multimodal lstm for speaker identification. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, 3581–3587.

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, 234–241. Springer.

Rössler, A.; Cozzolino, D.; Verdoliva, L.; Riess, C.; Thies, J.; and Nießner, M. 2018. Faceforensics: A large-scale video dataset for forgery detection in human faces. *arXiv*.

Sharma, A.; Grau, O.; and Fritz, M. 2016. Vconv-dae: Deep volumetric shape learning without object labels. In *European Conference on Computer Vision*, 236–250. Springer.

Simakov, D.; Caspi, Y.; Shechtman, E.; and Irani, M. 2008. Summarizing visual data using bidirectional similarity. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, 1–8. IEEE.

Varley, J.; DeChant, C.; Richardson, A.; Ruales, J.; and Allen, P. 2017. Shape completion enabled robotic grasping. In *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*, 2442–2447. IEEE.

Venkatesh, M. V.; Cheung, S.-c. S.; and Zhao, J. 2009. Efficient object-based video inpainting. *Pattern Recognition Letters* 30(2):168–179.

Wang, C.; Guo, Y.; Zhu, J.; Wang, L.; and Wang, W. 2014. Video object co-segmentation via subspace clustering and quadratic pseudo-boolean optimization in an mrf framework. *IEEE Transactions on Multimedia* 16(4):903–916.

Wang, C.; Zhu, J.; Guo, Y.; and Wang, W. 2017a. Video vectorization via tetrahedral remeshing. *IEEE Transactions on Image Processing* 26(4):1833–1844.

Wang, W.; Huang, Q.; You, S.; Yang, C.; and Neumann, U. 2017b. Shape inpainting using 3d generative adversarial network and recurrent convolutional networks. *arXiv preprint arXiv:1711.06375*.

Wexler, Y.; Shechtman, E.; and Irani, M. 2004. Space-time video completion. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, I–I. IEEE.

Wexler, Y.; Shechtman, E.; and Irani, M. 2007. Space-time completion of video. *IEEE Transactions on pattern analysis and machine intelligence* 29(3).

Xie, J.; Xu, L.; and Chen, E. 2012. Image denoising and inpainting with deep neural networks. In *Advances in neural information processing systems*, 341–349.

Yang, C.; Lu, X.; Lin, Z.; Shechtman, E.; Wang, O.; and Li, H. 2017. High-resolution image inpainting using multi-scale neural patch synthesis. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, 3.