Copy-and-Paste Networks for Deep Video Inpainting

Sungho Lee
Yonsei University

Seoung Wug Oh
Yonsei University

DaeYeun Won
Hyundai MNSOFT

Seon Joo Kim
Yonsei University

Abstract

We present a novel deep learning based algorithm for video inpainting. Video inpainting is a process of completing corrupted or missing regions in videos. Video inpainting has additional challenges compared to image inpainting due to the extra temporal information as well as the need for maintaining the temporal coherency. We propose a novel DNN-based framework called the Copy-and-Paste Networks for video inpainting that takes advantage of additional information in other frames of the video. The network is trained to copy corresponding contents in reference frames and paste them to fill the holes in the target frame. Our network also includes an alignment network that computes affine matrices between frames for the alignment, enabling the network to take information from more distant frames for robustness. Our method produces visually pleasing and temporally coherent results while running faster than the state-of-the-art optimization-based method. In addition, we extend our framework for enhancing over/under exposed frames in videos. Using this enhancement technique, we were able to significantly improve the lane detection accuracy on road videos.

1. Introduction

Inpainting is a task of completing an image that has empty pixels by filling the empty regions with visually plausible pixels. Inpainting is very useful in image editing process, and is usually utilized to generate more satisfying images by removing unwanted objects in images. There is a large body of literature on image inpainting and significant progress has been made recently by employing deep learning for image inpainting. Impressive inpainting results are reported by applying evolving deep generative models [7], synthesizing visually pleasing images even for complex scenes.

In this paper, we focus on the video inpainting problem. Videos with additional temporal information makes the already difficult problem even more challenging. In addition to filling the holes for every frame, the algorithm has to ensure that the completed frames are temporally consistent. Due to these challenges, we have only seen one work that tackles the problem using deep neural networks (DNN) [13], compared to the image inpainting problem where many deep learning based algorithms have been introduced.

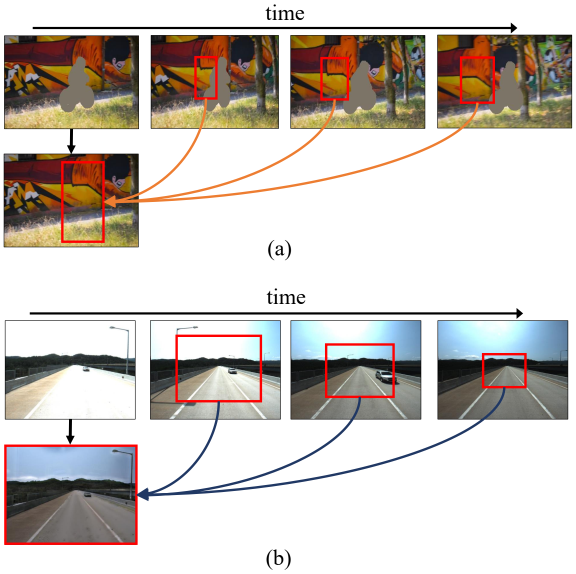


Figure 1: (a) We propose a DNN framework for video inpainting. Our Copy-and-Paste network learns to find corresponding pixels in other frames to fill in the holes in the given frame. (b) Another application of our framework for restoring an over-saturated image.

While video inpainting is more challenging compared to image inpainting, it inherently includes more cues for the problem as valid pixels for missing regions in a frame may exist in other frames. Therefore, we propose a novel DNN based framework called the *Copy-and-Paste Networks* for video inpainting that takes advantage of additional information in other frames in the video. As the name suggests, the network is trained to copy the necessary pixels from other frames and paste those pixels on the holes in the current frame (Fig. 1).

2380-7504/19/$31.00 ©2019 IEEE
DOI 10.1109/ICCV.2019.00451

The key components of our DNN system are the alignment and the context matching. To find corresponding pixels in other frames for the holes in the given frame, the frames need to be registered first. We propose a self-supervised alignment networks, which estimates affine matrices between frames. While DNNs for computing the affine matrix or homography exist [5, 11, 17], our alignment method is able to deal with holes in images when computing the affine matrices. After the alignment, the novel context matching algorithm is used to compute the similarity between the target frame and the reference frames. The network learns which pixels are valuable for copying through the context matching, and those pixels are used to paste and complete an image. By progressively updating the reference frames with the inpainted results at each step, the algorithm can produce videos with temporal consistency.

Our results are comparable to the state-of-the-art method [9], and outperform other deep learning based approaches [13, 24]. Moreover, we can easily extend our method for restoring saturated/under-exposed images as shown in (Fig. 1(b)). By enhancing the saturated/under-exposed images, we were able to significantly increase the lane detection accuracy.

In summary, the major contribution of our paper is as follows:

- We propose a self-supervised deep alignment networks that can compute affine matrices between images that contain large holes.
- We propose a novel context-matching algorithm to combine reference frame features based on similarity between images.
- Our method produces visually pleasing completed videos, running much faster than the state-of-the-art method. Additionally, we extend our framework for enhancing over/under exposed frames in videos that can help to improve other vision tasks such as the lane detection.

2. Related works

2.1. Image Inpainting

In traditional image inpainting methods, an image is filled by referencing pixels outside the hole in the image or in the external image database. As one of the most representative inpainting methods, PatchMatch [1] reconstructs the missing region by searching the patches outside the hole based on the approximate nearest neighbor algorithm. With this type of approach, however, it is difficult to inpaint images with complicated scenes, or when the images do not contain sufficient information for filling the holes.

Since deep image inpainting has been introduced in [10, 18], many deep generative models for image inpainting have been proposed recently, showing impressive restoration results on complex scenes. Yu *et al.* [24] proposed the contextual attention module between the completed structure of the hole area and the patches outside the hole. Liu *et al.* [15] and Yu *et al.* [23] applied the partial convolution and the gated convolution to compensate the weakness of the vanilla convolution for image inpainting. In particular, Liu *et al.* [15] corrected the blurred results based on the perceptual and the style loss without the adversarial loss.

2.2. Video Inpainting

Video inpainting has additional challenges of restoring the holes in every frame and maintaining the temporal consistency between reconstructed frames. Meanwhile, unlike in image inpainting, one can utilize redundant information between frames of video in video inpainting. However, directly exploiting the redundant information in videos is difficult due to image variation from the movements of the camera and the objects. To compensate for the movements, Granados *et al.* [8] proposed to align the frames based on the homographies. They also applied the optical flow between completed frames to maintain the temporal consistency.

In [16], Newson *et al.* proposed 3D PatchMatch to maintain the temporal consistency in addition to using the affine transformation to compensate the motion. While the spatio-temporal patches improve the short-term temporal consistency, the long-term consistency of complicated scenes remained as a limitation. To solve this limitation, Huang *et al.* [9] proposed the optical flow optimization in spatial patches to complete images while preserving the temporal consistency. This method shows the state-of-the-art performance up until now. All the methods explained above are based on a heavy optimization, and therefore suffers in the computational time, limiting their practical use.

Wang *et al.* [22] proposed the first deep learning based video inpainting by using 3D encoder-decoder networks. However, this work does not cover the object removal task in general videos, and was only applied to a few specific domains. Kim *et al.* [13] proposed 3D-2D encoder-decoder networks to complete the missing contents efficiently. The temporal consistency is maintained through a recurrent feedback and a memory layer with the flow and the warping loss. The temporal window for the referencing is small in their method, and therefore it is difficult to use valid pixels in distant frames, resulting in a limited performance for scenes with large objects or slowly moving objects.

Our copy-and-paste network overcome the issues in [13] by aligning the frames with affine matrices computed by our alignment network instead of using the optical flow. With the novel context matching algorithm, our method can extract valid pixels in distant frames, resulting in more ac-

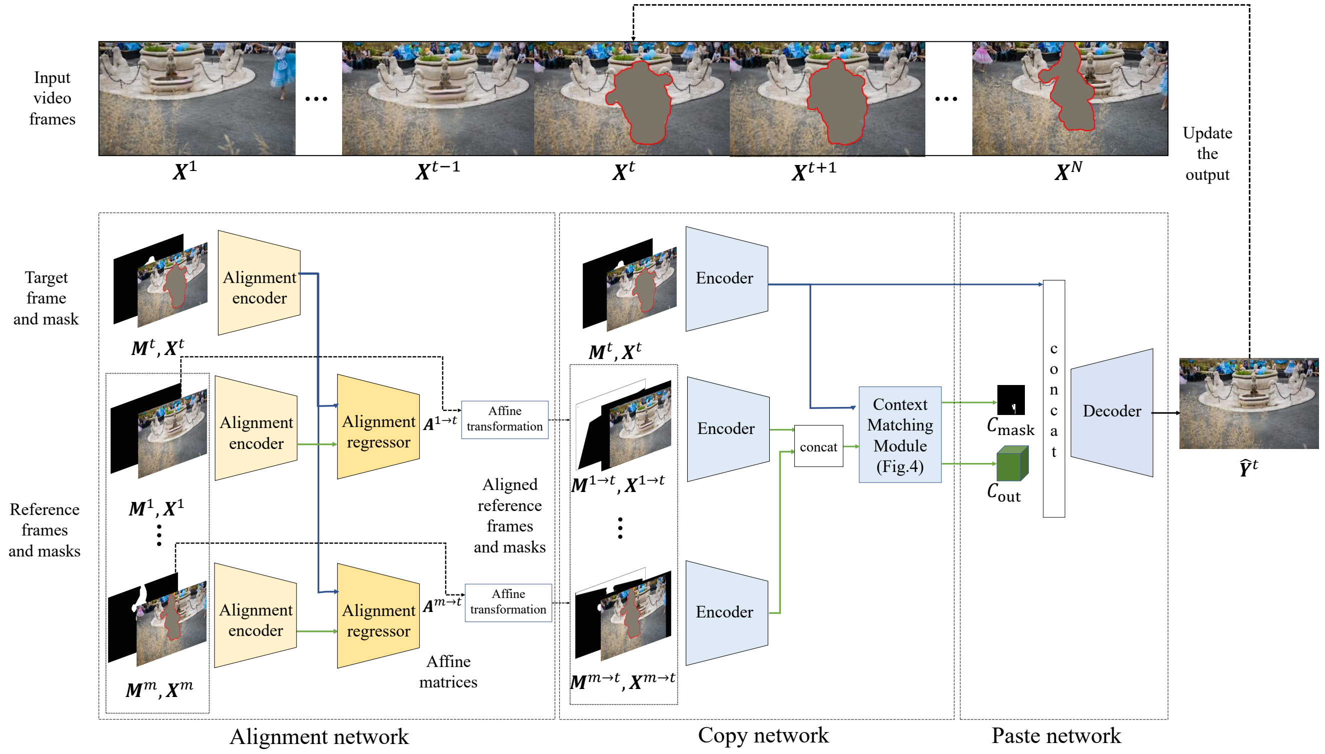


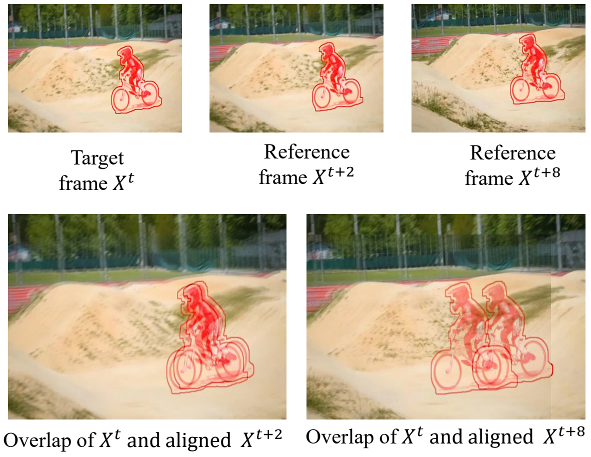
Figure 2: Network Overview. Our framework consists of 3 sub-networks: alignment network, copy network, and paste network.

curate reconstruction for general scenes. The performance of our method is comparable to the state-of-the-art method in [9] while being more practical with faster runtime due to the feed forward nature of DNNs.

3. Copy-and-Paste Network Algorithm

The overview of our framework is shown in Fig. 2. The system takes a video ($\boldsymbol{X}$) annotated with the missing pixels ($\boldsymbol{M}$) in each frame and outputs ($\hat{\boldsymbol{Y}}$) the completed video. The video is processed frame-by-frame in the temporal order. We call the frame to be filled as the target frame and the other frames as the reference frames. For each target frame, our network completes the missing region by copying-and-pasting contents from the reference frames.

To complete a target frame, each reference frame is first aligned to the target frame through the alignment network. Then in the copy network, pixels to be copied from the aligned reference frames are determined by the context matching module. Finally, the outputs from the copy networks are decoded to produce inpainted target frame in the paste network. The input video in the memory is updated with the completed frame, which will subsequently be used as a reference frame, providing more information for the following frames.


Target frame X^t

Reference frame X^{t+2}

Reference frame X^{t+8}

Overlap of X^t and aligned X^{t+2}

Overlap of X^t and aligned X^{t+8}

Figure 3: Using affine transformation for the alignment yields larger temporal search compared to the optical flow based alignment. More distant reference frame provides more valuable information as the overlap of the hole regions is smaller.

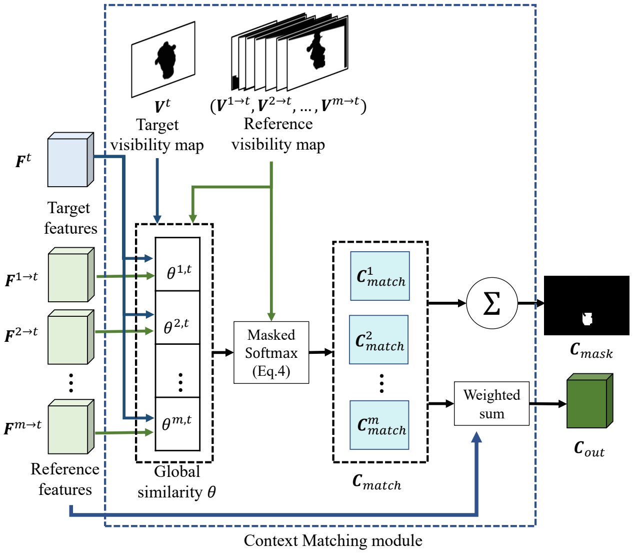


Figure 4: Detailed illustration of the context matching module.

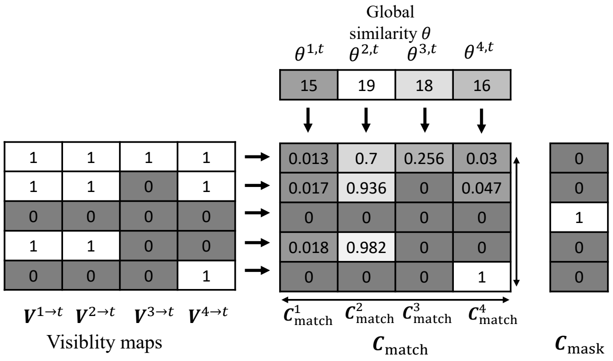
3.1. Alignment Network

In video inpainting, a large temporal window is essential as valuable information is more likely to be in distant frames. With an optical flow based alignment as used in [13], the temporal range of information is too small to extract useful information. As illustrated in Fig. 3, a reference frame temporally close to the target frame lacks information to fill the hole as there are too much overlap between the holes in the images. Moreover, computing optical flows between images with holes is more difficult as the holes themselves become occlusion factors. Therefore, our alignment network estimates the affine matrices to align the reference frames with the target frame.

The alignment network consists of shared alignment encoders and alignment regressors. Details on the network architectures are provided in the supplementary materials. To train the alignment network, we minimize the self-supervised loss, which is the L1 distance between the target frame ($\boldsymbol{X}^t$) and the aligned reference frame ($\boldsymbol{X}^{r\to t}$). To exclude the hole regions, this pixel-wise loss is only measured with pixels that are valid in both images as follows:

$$\mathcal{L}_{\text{align}} = \sum_r ||\boldsymbol{V} \odot (\boldsymbol{X}^t - \boldsymbol{X}^{r\to t})||_1, \tag{1}$$

where $\boldsymbol{V} = \boldsymbol{V}^t \odot \boldsymbol{V}^{r\to t}$ is the visibility map, $\odot$ is the element-wise product, t is the target frame index, and r is the reference frame index[1]. The visibility map is computed



$\theta^{1,t}$	$\theta^{2,t}$	$\theta^{3,t}$	$\theta^{4,t}$
15	19	18	16

$V^{1\to t}$	$V^{2\to t}$	$V^{3\to t}$	$V^{4\to t}$
1	1	1	1
1	1	0	1
0	0	0	0
1	1	0	0
0	0	0	1

C^1_{match}	C^2_{match}	C^3_{match}	C^4_{match}	C_{mask}
0.013	0.7	0.256	0.03	0
0.017	0.936	0	0.047	0
0	0	0	0	1
0.018	0.982	0	0	0
0	0	0	1	0

Figure 5: An 1-D example of masked softmax.

from the given masks, where 0 indicates hole pixels and 1 represents non-hole pixels. Note that the alignment network is jointly trained with other networks in an end-to-end manner, not independently.

3.2. Copy-and-Paste Network

After the frame alignment, the aligned frames are mapped into the feature space through the shared encoders. The context matching module computes the importance of each pixel in the reference frames in completing the holes as well as a mask ($\boldsymbol{C}_{\text{mask}}$) indicating the visibility of each pixel throughout the video. Finally, the decoder takes the output of the context matching module in addition to the target frame feature to restore values for the missing pixels.

Encoder Encoder networks extract the features from the target and the aligned reference frames. The input to the encoder is a concatenation of an RGB image and the corresponding binary mask. The details on the architecture will be described in the supplementary materials.

Context matching module Together with the encoder, the context matching module constitutes the copy network. The context matching module is illustrated in Fig. 4. First, global similarities ($\theta^{r,t}$) between the aligned reference frames and the target frame in the feature space is computed as follows:

$$\theta^{r,t} = \frac{1}{\sum_{(x,y)} \boldsymbol{V}(x,y)} \cdot \sum_{(x,y)} \boldsymbol{V}(x,y) \cdot \boldsymbol{F}^t(x,y) \cdot \boldsymbol{F}^{r\to t}(x,y). \tag{2}$$

The above equation is basically computing the cosine similarity between the two feature maps, excluding the hole pixels.

Then, a saliency map $\boldsymbol{C}^r_{\text{match}}$ for each reference frame is computed as follows:

$$\boldsymbol{S}^{r,t} = \theta^{r,t} \cdot \boldsymbol{V}^{r\to t}, \tag{3}$$

[1] The symbol $r \to t$ indicates aligning a reference frame r to a target frame t. $\boldsymbol{V}^{r\to t}$ indicates the visibility map of the reference aligned to the target

$$\boldsymbol{C}^r_{\text{match}}(x,y) = \begin{cases} \frac{exp(\boldsymbol{S}^{r,t}(x,y))}{\sum_r exp(\boldsymbol{S}^{r,t}(x,y))} & \text{if } \boldsymbol{V}^{r \to t}(x,y) = 1 \\ 0 & otherwise. \end{cases} \tag{4}$$

Fig. 5 simplifies the steps for computing the saliency map in 1-D. Each pixel value in the saliency map $\boldsymbol{C}^r_{\text{match}}$ holds the weight that specific pixels have on filling the hole in the target. The reference features are aggregated through a weighted sum with the $\boldsymbol{C}^r_{\text{match}}$, producing the features to be used for the decoder (C_{out}).

$$\boldsymbol{C}_{\text{out}}(x,y) = \sum_r \boldsymbol{F}^{r \to t}(x,y) \cdot \boldsymbol{C}^r_{\text{match}}(x,y). \tag{5}$$

The hole masks for the reference frames are also aggregated in a similar fashion, resulting in $\boldsymbol{C}_{\text{mask}}$. $\boldsymbol{C}_{\text{mask}}$ indicates pixels that is never visible throughout the reference frame.

The process of the aggregation is expressed as:

$$\boldsymbol{C}_{\text{mask}}(x,y) = 1 - (\sum_r \boldsymbol{C}^r_{\text{match}}(x,y)). \tag{6}$$

Decoder The decoder network completes the target frame given target features, aggregated reference features, and mask C_{mask}. The inputs are concatenated before being fed into the decoder. Decoder is basically our paste network that learns to fill the missing region by using the aggregated reference features and the visibility of those features. The pixels marked on C_{mask} are pixels that are never visible in all reference frame because those pixels always fall into holes. Therefore, the decoder has to be able to synthesize contents for those pixels as well. We add dilated convolution blocks to grow the receptive field and design the decoder network deeper than the other networks, in order to enhance the completion results for the unseen area by looking at other pixels within the image itself.

3.3. Temporal Consistency

Each frame in the video is sequentially completed by the network, one by one. The completed frame at each iteration replaces its reference, providing more information for the following frames as the holes are now filled with contents. This iterative reference update procedure not only improves the quality of the restored images, but also enhances the temporal consistency. This is analyzed later in the ablation study. To further ensure the temporal consistency, we actually run the feed-forward network twice – completing the video from the first to the last frame, and also in the reverse order. Then the final results are computed as follows:

$$\hat{\boldsymbol{Y}}^t_{\text{final}} = \hat{\boldsymbol{Y}}^t_{\text{forward}} \cdot \frac{t}{N} + \hat{\boldsymbol{Y}}^t_{\text{reverse}} \cdot \frac{(N-t)}{N}. \tag{7}$$

4. Training

4.1. Loss functions

All the networks are trained jointly in an end-to-end manner. First, we compute the loss between the completed target frame and the ground truth. The losses for the hole region and the non-hole region are separately calculated. Furthermore, the hole region can be divided into areas depending on whether the pixel value can be copied from reference frames or not. Therefore, we also apply the losses in the hole region separately.

$$\begin{aligned} \mathcal{L}_{\text{hole(visible)}} &= \sum_t^N \boldsymbol{M}^t \odot \boldsymbol{C}_{\text{mask}} \odot ||\hat{\boldsymbol{Y}}^t - \boldsymbol{Y}^t||_1, \\ \mathcal{L}_{\text{hole(invisible)}} &= \sum_t^N \boldsymbol{M}^t \odot (1. - \boldsymbol{C}_{\text{mask}}) \odot ||\hat{\boldsymbol{Y}}^t - \boldsymbol{Y}^t||_1, \\ \mathcal{L}_{\text{non-hole}} &= \sum_t^N (1 - \boldsymbol{M}^t) \odot ||\hat{\boldsymbol{Y}}^t - \boldsymbol{Y}^t||_1. \end{aligned} \tag{8}$$

$\boldsymbol{C}_{\text{mask}}$ is properly resized to fit the size of the target frame.

To further improve the visual quality of the results, we also apply perceptual, style, and total variation loss.

$$\begin{aligned} \mathcal{L}_{\text{perceptual}} &= \frac{1}{P} \cdot \sum_p^P ||\phi_p(\hat{\boldsymbol{Y}}_{\text{comp}}) - \phi_p(\boldsymbol{Y})||_1, \\ \mathcal{L}_{\text{style}} &= \frac{1}{P} \cdot \sum_p^P ||G^\phi_p(\hat{\boldsymbol{Y}}_{\text{comp}}) - G^\phi_p(\boldsymbol{Y})||_1, \end{aligned} \tag{9}$$

where $\hat{\boldsymbol{Y}}_{\text{comp}}$ is combination of the decoder output $\hat{\boldsymbol{Y}}^t$ in the hole region and the input $\boldsymbol{X}^t$ outside the hole, ϕ is the output of the pooling layer in pretrained VGG-16 [21] on ImageNet [4], p is the pooling index, G is the gram matrix multiplication [12].

The total-loss function is as follows:

$$\begin{aligned} \mathcal{L} = {} & 2 \cdot \mathcal{L}_{\text{align}} + 10 \cdot \mathcal{L}_{\text{hole(visible)}} + 20 \cdot \mathcal{L}_{\text{hole(invisible)}} \\ & + 6 \cdot \mathcal{L}_{\text{non-hole}} + 0.01 \cdot \mathcal{L}_{\text{perceptual}} + 24 \cdot \mathcal{L}_{\text{style}} + 0.1 \cdot \mathcal{L}_{\text{tv}}, \end{aligned} \tag{10}$$

where $\mathcal{L}_{\text{tv}}$ is the total variation loss for smoothing the checkerboard effect [12]. The weight for each loss is empirically determined.

4.2. Datasets

Our goal is to complete holes in video sequences. Inputs are image sequences with holes and binary masks indicating the hole regions. However, no public video dataset for video inpainting exist. Therefore, we synthesized a dataset for video inpainting using background images and segmentation masks.

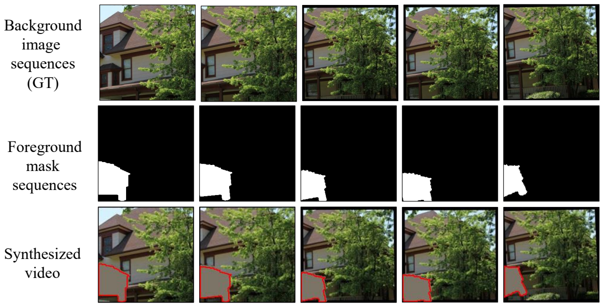


Figure 6: Synthesized training dataset example.

We synthesize videos by compositing background image sequences with object masks (Fig. 6). To build background image sequences, we use the Places (amount of 1.8M images) [25] single image datasets. To synthesize a sequence of images from a single image, we applied random crops and successive random transformations (shear, scale, translation, rotation) on the image. Additionally, we crawled the Youtube video clips and divided them according to the scene (7.3K scenes). Frames are randomly sampled from video clips to form a image sequence. The source of the background image sequence is randomly selected in an equal chance.

To simulate masks for holes, we use object masks from MIT Saliency Benchmark(amount of 11K masks) [2] and Pascal VOC 2012(amount of 14.3K masks) [6]. A mask is randomly resized to be smaller than the size of the background frames. And the mask is randomly transformed to be a mask sequence by simulating the moving objects. A training sample is made by compositing a background image sequence and a mask sequence made above.

4.3. Training Details

Our model runs on hardware with the Intel(R) Core(TM) i7-7800X CPU(3.50GHz) CPU and NVIDIA TITAN XP GPUs. We train with the randomly selected five 256 × 256 frames from the synthesized video sequences as inputs. To train the network, we set the batch size as 40. We use the Adam Optimizer [14] with learning rates 10^{-4} and reduce the running rate factor of 10 every 1 million iterations. The training process takes about 7 days using three NVIDIA TITAN XP GPUs.

5. Experiments

To evaluate our algorithm, we provide both quantitative and qualitative analysis, as well as a user study. We conducted the experiments using the videos, which were scaled in half (424 × 240). Our code will be available online. We also show an application of our work in restoring under/over-exposed images.

Method	PSNR	SSIM
Huang *et al.* [9]	28.14	0.859
Ours	28.37	0.851

Table 1: Quantitative Results (video restoration) for DAVIS 2017

5.1. Quantitative Results

We first conducted quantitative evaluation by measuring the quality of video restoration. For this experiment, we randomly selected 25 video sequences in DAVIS dataset [19, 20], which consists of pairs of video and object segmentation mask sequences. To simulate image restoration, we synthesized videos by putting imaginary object masks from DAVIS [19, 20] on the videos. The video without the object masks are used as the ground truth. Table 1 compares the PSNR and the SSIM measures between our method and [9]. Both methods show good performance with similar measures. Note that VINet [13] is excluded in this experiment because the official code has not been published yet.

5.2. User Study and Qualitative Analysis

We further conducted experiments on dynamic object removal in videos with 30 videos from DAVIS dataset [19, 20]. We compared our methods with the state-of-the-art video inpainting models [9, 13]. Results of the previous methods were gathered by using the official code released by the authors [9] and by requesting the results from the authors [13].

The user study result performed the Amazon Mechanical Turk (AMT) is shown in Fig. 8 and Table 2 . The workers were asked to rank the video completion results and we also allowed them to give ties. All tests were evaluated by 40 participants.

Method	Average ranking
Huang *et al.* [9]	1.74
VINet [13]	2.08
Ours	1.77

Table 2: User study average rank (lower value is better)

The user study shows that our method is highly competitive to the optimization based method [9], while VINet [13] is not on par with the other two methods. While the method in [9] was slightly more favored, it requires average completion time of 952 seconds per video, whereas our method only takes **27.14** seconds.

Qualitative comparisons of the object removal results are shown in Fig. 7. These comparisons show similar results as the user study. Our results are comparable to the state-

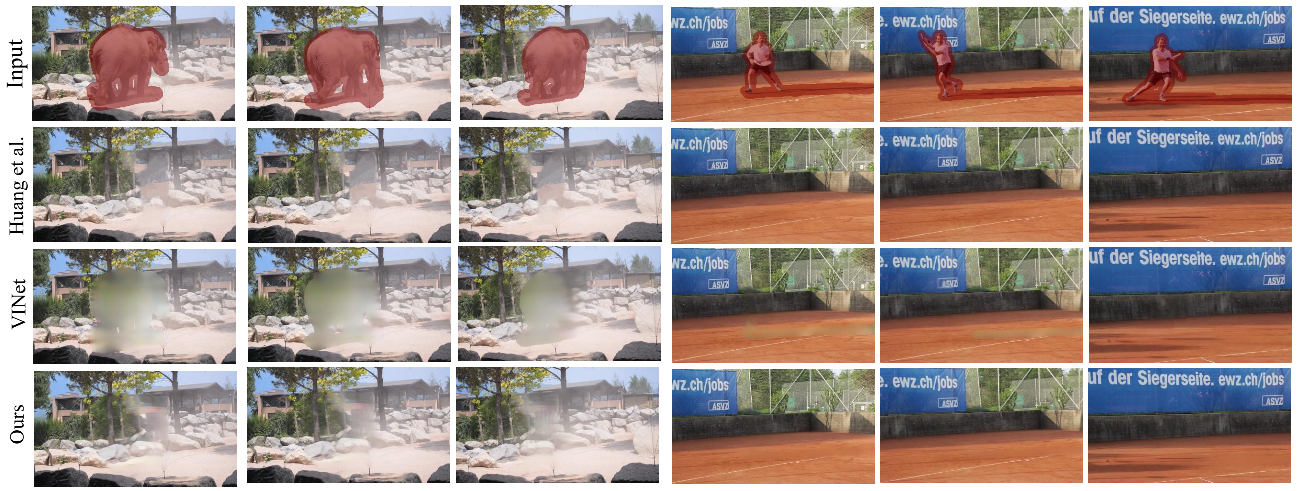


Figure 7: Qualitative comparison of object removal results for the scenes *elephant* (left) and *tennis* (right) from DAVIS 2017 sequences.

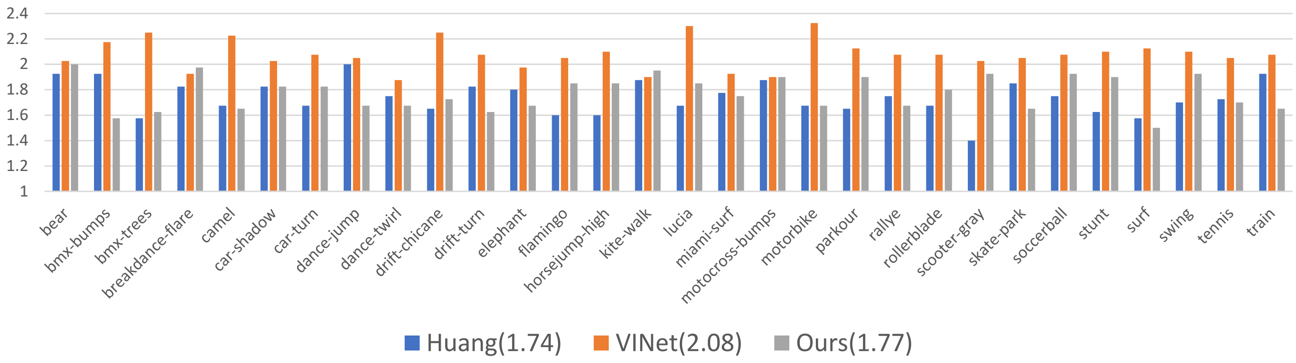


Figure 8: User study for video object removal results (lower value is better)

of-the-art method in [9], while showing much better results compared to the other deep learning based approach in [13].

5.3. Applications

We extend our method for restoring under/over-exposed image sequences. The restoration process is similar to video inpainting problem in that it fills areas with missing information. This problem often happens to image sequences taken by a camera attached to a vehicle due to rapid exposure changes (*e.g*. tunnel entry and exit).

As shown in Fig. 9, both the texture and the color are improved. To validate the effectiveness of our restoration process, we ran a lane detection algorithm on road images before and after the enhancement. We collected 469 frames videos [2] that contains rapid exposure changes due to tunnels and the internal color histogram-based lane detection method was used. As shown in Fig. 9 and Table 3, lane detection results are significantly improved.

Lane detection input	Lane detection accuracy
Over/under-exposed image	46.69%
Restored input by our model	**83.00%**

Table 3: The lane detection accuracy results.

6. Ablation Study

Masked softmax We conducted an ablation study to verify that masked softmax contributes to the performance improvements. We train our model using normal softmax under the same conditions. As shown in the Fig. 10, using masked softmax results are sharper than using the normal one.

[2] The dataset were taken by using Mobile Mapping System Camera of Hyundai MnSOFT, Inc.

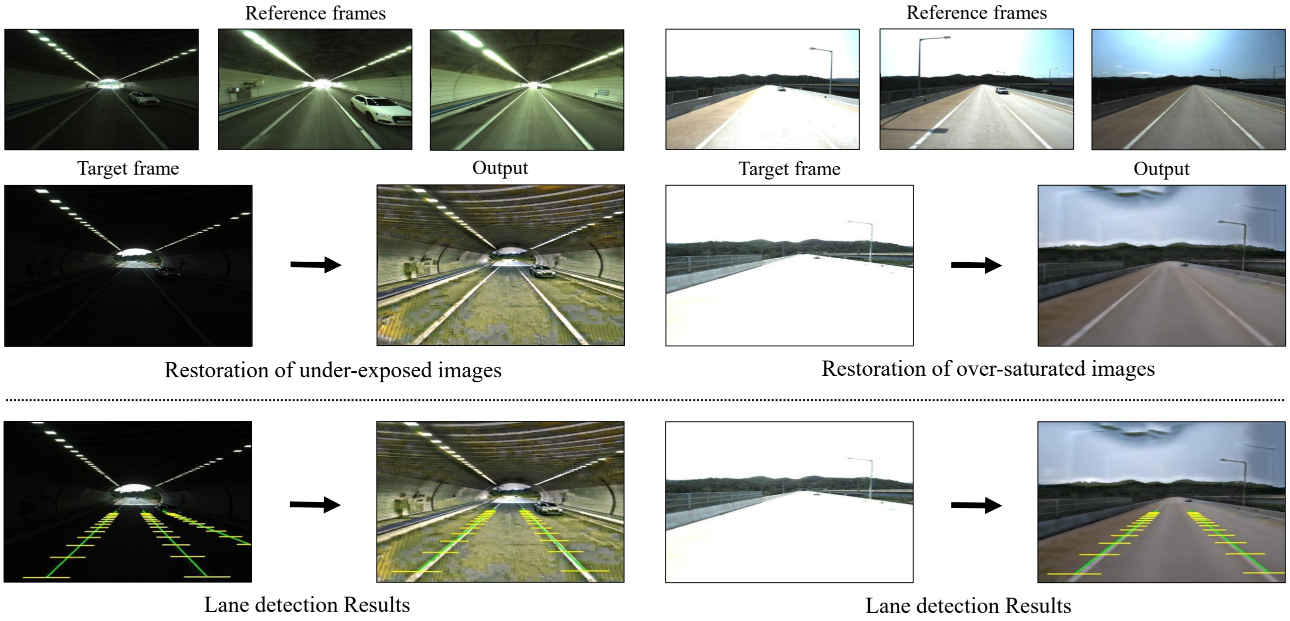
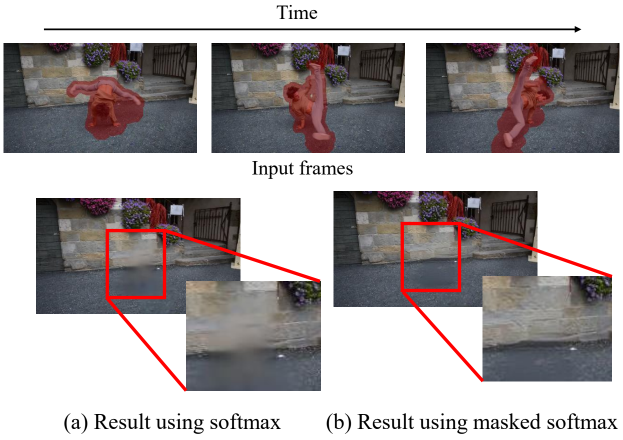
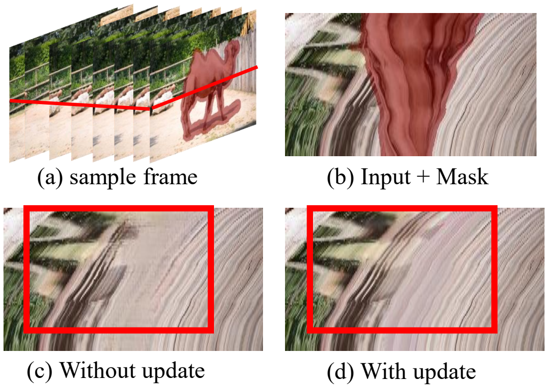


Figure 9: Application of our method for the restoration of under/over-exposed images.



(a) Result using softmax (b) Result using masked softmax

Figure 10: Ablation study for masked softmax.



(a) sample frame (b) Input + Mask

(c) Without update (d) With update

Figure 11: Ablation study for reference update. (b), (c) and (d) show the temporal profile of the red line shown in input (a).

Reference update To produce temporally coherent outputs, we update the past reference frames with the inpainted version. To visualize the effect of this updating protocol, we compare the temporal profile [3] of resulting videos in Fig. 11. As shown in Fig. 11, the update procedure contributes in enhancing the temporal consistency.

7. Conclusion

In this paper, we presented a novel DNN framework for video inpainting. The proposed method inpaints the missing information by copy-and-pasting contents from the reference frames. The reference information is dynamically updated by the previous completion results to ensure the temporal consistency. Our experiments support that the proposed framework is comparable to the optimization-based methods and outperform other deep learning based approaches. We extended our framework to restore over/under-exposed in videos and were able to significantly increase the lane detection accuracy.

Acknowledgement

This work was supported by Institute for Information & communication Technology Promotion (IITP) grant funded by the Korea government (MSIP) (2018-0-01858).

References

[1] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. Patchmatch: a randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics (TOG)*, 28(3):24, 2009. 2

[2] Zoya Bylinskii, Tilke Judd, Ali Borji, Laurent Itti, Frédo Durand, Aude Oliva, and Antonio Torralba. Mit saliency benchmark. 6

[3] Jose Caballero, Christian Ledig, Andrew Aitken, Alejandro Acosta, Johannes Totz, Zehan Wang, and Wenzhe Shi. Real-time video super-resolution with spatio-temporal networks and motion compensation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4778–4787, 2017. 8

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 5

[5] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Deep image homography estimation. *arXiv preprint arXiv:1606.03798*, 2016. 2

[6] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111(1):98–136, 2015. 6

[7] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 1

[8] Miguel Granados, Kwang In Kim, James Tompkin, Jan Kautz, and Christian Theobalt. Background inpainting for videos with dynamic objects and a free-moving camera. In *ECCV*, 2012. 2

[9] Jia-Bin Huang, Sing Bing Kang, Narendra Ahuja, and Johannes Kopf. Temporally coherent completion of dynamic video. *ACM Transactions on Graphics (TOG)*, 35(6), 2016. 2, 3, 6, 7

[10] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Globally and Locally Consistent Image Completion. *ACM Transactions on Graphics (Proc. of SIGGRAPH 2017)*, 36(4):107:1–107:14, 2017. 2

[11] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in neural information processing systems*, pages 2017–2025, 2015. 2

[12] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016. 5

[13] Dahun Kim, Sanghyun Woo, Joon-Young Lee, and In So Kweon. Deep video inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5792–5801, 2019. 1, 2, 4, 6, 7

[14] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[15] Guilin Liu, Fitsum A Reda, Kevin J Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 85–100, 2018. 2

[16] Alasdair Newson, Andrs Almansa, Matthieu Fradet, Yann Gousseau, and Patrick Prez. Video inpainting of complex scenes. *SIAM Journal on Imaging Sciences, Society for Industrial and Applied Mathematics*, 7(4):1993–2019, 2014. 2

[17] Ty Nguyen, Steven W Chen, Shreyas S Shivakumar, Camillo Jose Taylor, and Vijay Kumar. Unsupervised deep homography: A fast and robust homography estimation model. *IEEE Robotics and Automation Letters*, 3(3):2346–2353, 2018. 2

[18] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. Context encoders: Feature learning by inpainting. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[19] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 724–732, 2016. 6

[20] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017. 6

[21] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 5

[22] 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 Intelligence*, volume 33, pages 5232–5239, 2019. 2

[23] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. *arXiv preprint arXiv:1806.03589*, 2018. 2

[24] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5505–5514, 2018. 2

[25] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2017. 6