

# An Internal Learning Approach to Video Inpainting

## Supplementary Material

### Abstract

*In this supplementary material, we provide more details about our network architecture and model training. The source code can be found at [https://github.com/Haotianz94/IL\\_video\\_inpainting](https://github.com/Haotianz94/IL_video_inpainting).*

## 1. Network Architecture

In our experiments, we use two different networks. Our 2D baselines (DIP and DIP-Vid) and our final model (DIP-Vid-Flow) share the same Encoder-Decoder architecture. Our 3D baseline (DIP-Vid-3DCN) uses a modified version with 3D convolution.

### 1.1. 2D Network

**Encoder** The Encoder consists of 12 convolution layers. Every two consecutive layers form a block, where the two layers have the same number of channels. The first layer in each block uses the stride of 2 to reduce the spatial resolution. All the convolution layers use the filter size of 5. The number of channels for each layer is shown below. C16-C16-C32-C32-C64-C64-C128-C128-C128-C128-C128-C128

**Decoder** The Decoder also consists of 12 convolution layers in 6 blocks. One Nearest-neighbor upsampling layer is added to the beginning of each block. All the convolution layers use the filter size of 3. The number of channels for each layer is shown below. (Not fully symmetric to the Encoder due to the skip connections) C132-C128-C132-C128-C132-C128-C68-C64-C36-C32-C20-C16

**Skip Connection** A skip connection is added from the beginning of the  $i$  th block of the Encoder to the beginning of the  $(n - i)$  th block (after the upsampling layer) of the Decoder. All skip connections use one convolution layer with 4 channels and filter size of 1.

**Final Layer** For DIP and DIP-Vid, the final layer only contains a convolution layer with 3 channels followed by a sigmoid to generate the final image. For DIP-Vid-Flow, an

other flow generation branch is added parallel to the image generation branch. The flow generation branch contains a convolution layer with 12 channels, corresponding to 6 different flow maps of temporal range 1, 3, 5 in both forward and backward directions.

All the convolution layers except those in the final layer is followed by a Batch-Norm layer and a LeakyReLU layer with slope 0.2.

### 1.2. 3D Network

Our 3D version of the network shares the same structure with the 2D version except all the 2D convolution layers are replaced with 3D convolution layers. We also keep all the number of channels and filter size as the same as our 2D version. For the added  $3^{rd}$  dimension, we use the filter size of 3 for the Encoder and the Decoder and the filter size of 1 for the skip connections.

## 2. Network Input

As mentioned in the main paper, we sample the input noise maps independently for each frame and fix them during training. The noise map only has one channel and shares the same spatial size with the input frame. Each noise map is filled with uniform noise between 0 and 0.1. For our 2D network, we feed input noise maps as a 2D batch of dimension  $N \times 1 \times H \times W$ , where  $N$  is the batch size. For our 3D network, we directly transfer the 2D batch into a 3D batch of dimension  $1 \times 1 \times N \times H \times W$ , where  $N$  becomes the size of the  $3^{rd}$  dimension with batch size of 1. In all of our experiments, frames are resized and cropped to the resolution of  $384 \times 192$ .

## 3. Network Training

In this section, we describe the training details for all the baselines and our final model.

### 3.1. DIP

We train a DIP model for each frame independently. Due to the destabilization issue mentioned in the original paper [1], we run optimization on each frame for 5k iterations and

save the result every 100 iterations. The intermediate result with the lowest loss is chosen as the final inpainting result.

### 3.2. DIP-Vid

We train a single DIP model on the entire video. In each epoch, we randomly pick  $N$  consecutive frames as a training batch to enumerate all the possible batch permutations. Inspired by the training procedure used in DIP, we run optimization on the selected batch for  $M$  iterations before moving to the next batch. After training for  $E$  epochs, we run one inference using the trained model to get the final inpainting results. Only the image generation loss is applied in this baseline. The destabilization issue is also observed in this method, but considerably rare compared to DIP.

### 3.3. DIP-Vid-3DCN

All the settings are as the same as DIP-Vid, except for replacing the 2D network with the 3D version.

### 3.4. DIP-Vid-Flow

In our final model, we need to generate both images and flows. We randomly pick  $N$  frames which are consecutive with a fixed frame interval of  $t$  as a batch,  $t \in \{1, 3, 5\}$ . We do not use intervals larger than 5 due to the increasing error in estimated flows. We run optimization on the batch with all the image and flow related loss (See Sec3.1 in our main paper), but only using forward or backward flow at interval  $t$  for  $M$  iterations. Optimizing flows in both directions at the same time is observed to cause artifacts in the results occasionally, potentially due to the conflict in the flows. We select batches by enumerating all the possible permutations and finish training after  $E$  epochs on the whole video.

In all of our experiments, we use  $N = 5$ ,  $M = 100$  and  $E = 20$ .

## References

- [1] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9446–9454, 2018. 1