The DEVIL is in the Details: A Diagnostic Evaluation Benchmark for Video Inpainting Supplementary Materials

1. DEVIL Dataset Details

1.1. Source Videos

To identify a high-quality set of source videos depicting scenic landscapes, we begin by searching Flickr [7] for videos that contain the term "scenic" in their metadata. From these preliminary results, we identify a small number of users who upload a large volume of high-quality, non-post-processed videos. We then refine our search to "scenic" videos from those users within a given upload time frame (January 2017 - January 2019). From these videos, we automatically detect and discard any that contain shot transitions, resolutions not equal to 1920×1080, or COCO object classes [9] as detected by a Mask R-CNN model [5] provided by Detectron2 [12]. After automatic filtering, we manually inspect the remaining videos and remove those that contain undetected foreground objects or shot transitions, as well as other signs of post-processing (*e.g.*, sped-up videos). We split the remaining videos into clips containing between 45-90 frames, which constitute a grand total of 1,250 source clips.

1.2. Source Video Attributes

To annotate high BG scene motion, we manually identify clips that contain running bodies of water that cover at least 40% of the frame for all frames; for low BG scene motion, we identify clips that contain no running bodies of water (we establish our BG scene motion annotations based on bodies of water since they are prevalent in our data and easy for people to identify by visual inspection). We did not use automatic classifiers for this attribute due to their poor performance and the automatic bias that would have been introduced through their usage.

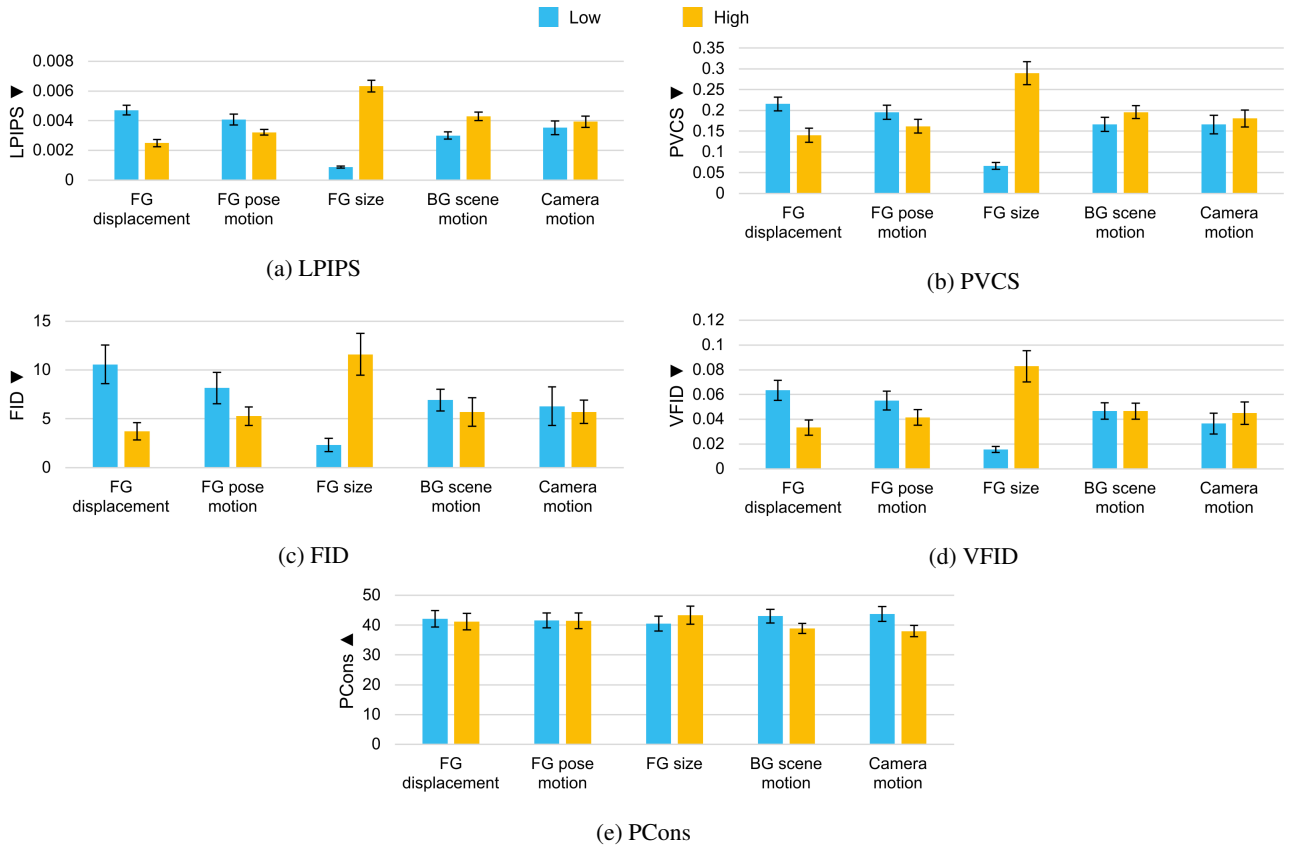
To annotate camera motion, we use classical affine alignment techniques and measure the amount of invalid pixels introduced via warping as a proxy for camera motion. The intuition behind this classifier is that high camera motion produces frames with poor pairwise affine alignments, and that warping frames by such transforms introduces a high percentage of invalid pixels into the field of view (the converse is true for low camera motion). Despite the simplicity of this approach, we found that it achieves a sufficiently high precision-recall AUC for our purposes (0.90 on a manually-annotated version of the DAVIS train/val set [10]).

Concretely, we label camera motion as follows: between a given pair of video frames, we first compute bidirectional robust affine transformations using RANSAC [4] over matched SURF keypoints [1]. Then, we warp the frames by the corresponding affine transformation and compute the number of invalid pixels introduced by the warp; we define the inverse of this quantity as the pairwise compatibility between the given frames. For a given clip, we sample ten evenly-spaced frames and compute the minimum pairwise compatibility between all pairs, which we define as the total frame compatibility of the clip. Finally, we obtain camera motion annotations by thresholding the total frame compatibility.

1.3. Occlusion Masks

To generate occlusion masks with our desired DEVIL attributes, we opt for a procedural generation approach inspired by Chang *et al*. [3], which enables fine-grained control over mask shape and behavior. In their framework, an initial mask shape is generated by sampling control points along a random walk with momentum (*i.e.*, biased toward an initial direction), and then connecting the control points with a stroke of random thickness. The mask is animated by moving all control points with a given velocity and then slightly perturbing their positions at each time step.

We extend the code of Chang *et al*. with several changes to enable even finer control over mask size and motion. For example, we reduce the impact of momentum in the initial mask-drawing phase to increase the diversity of mask shapes.



(a) LPIPS

(b) PVCS

(c) FID

(d) VFID

(e) PCons

Figure 1: Comparison of DEVIL slice difficulty. ▼ and ▲ indicate that lower and higher is better, respectively. Error bars show standard error across the seven evaluated methods.

Additionally, we apply inward-facing acceleration to the control points whenever they are sufficiently far from the mask's centroid, which effectively constrains its maximum possible area. Furthermore, we force the control points to bounce off the edge of the frame to prevent them from leaving the field of view. Finally, we randomly reverse the temporal dimension of masks with a 50% probability since they tend to grow in size over time.

Because the mask generation procedure is parameterized, we can produce occlusion masks that correspond to our desired DEVIL attribute settings by sampling from distinct configurations. To generate masks with small and large FG sizes, we sample from two corresponding ranges of stroke widths, and also change the maximum possible distance of each control point to the centroid. To vary the FG displacement, we sample the initial velocity of the overall mask from two different ranges. Finally, to generate masks with low and high pose motion, we vary the stochasticity of the control points (*i.e.*, for low pose motion masks, the control points are less likely to accelerate in a random direction per frame).

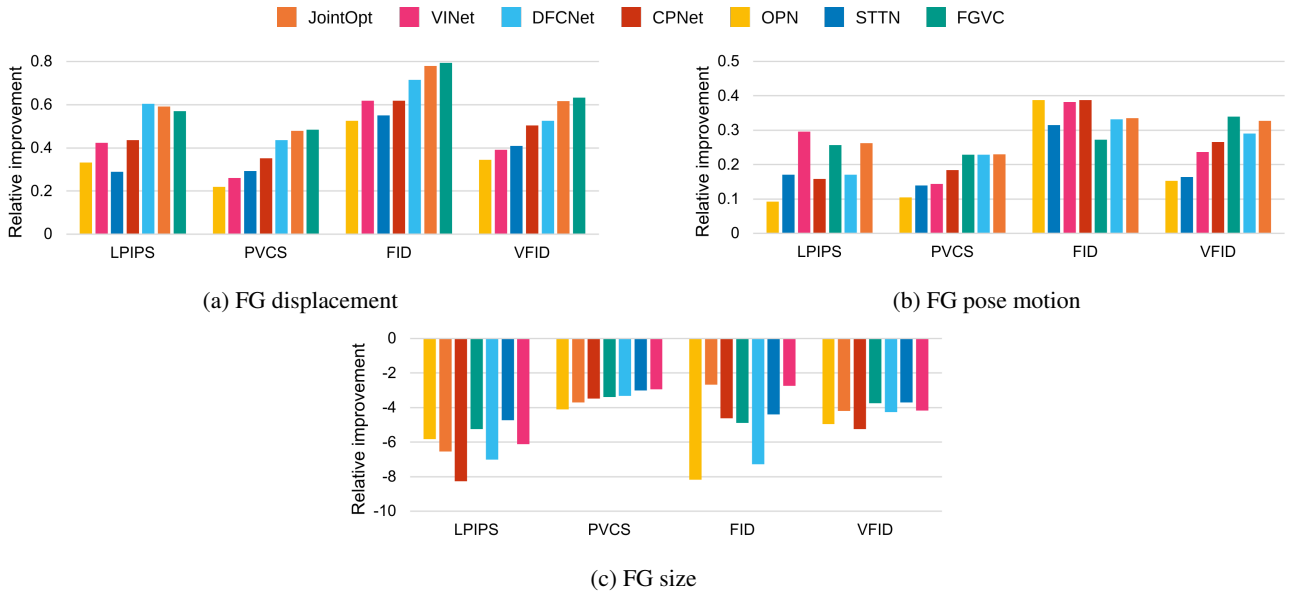
2. Evaluation Metric Details

In this section, we further describe our evaluation metrics, including details on the features used for the deep neural network-based metrics and parameters for the temporal consistency metric.

LPIPS and PVCS Our implementation of LPIPS is derived from the original code from Zhang *et al.* [13]. We use their fine-tuned AlexNet model weights [8] as well as their feature activations. For PVCS, we extend the LPIPS code to use a pre-trained I3D model [2] in place of the AlexNet model; distance is computed from the feature activations from I3D's five pre-pooling layers.

FID and VFID Our implementation of FID is derived from a third-party implementation of the metric from Heusel *et al.* [6].[1] The representation of a video frame corresponds to the activations from the final pooling layer of the Inception

[1]The third-party implementation is available at https://github.com/mseitzer/pytorch-fid.



(a) FG displacement

(b) FG pose motion

(c) FG size

Figure 2: Relative improvement of each method under reconstruction and realism metrics when DEVIL mask attributes change from low to high. Within each plot, the methods are sorted by PVCS performance.

Network [11], followed by global mean pooling over the remaining spatial dimensions. For VFID, we extend the third-party implementation of FID to use the same pre-trained I3D model as PVCS. To obtain the representation of a video, we extract the activations of I3D's final pooling layer and compute the average over the spatial and temporal dimensions (VFID is thus the Fréchet distance over video representations).

PCons To compute PCons between two frames, we first extract the 50×50 patch centered at the centroid of the mask from the first frame (if this patch partially lies beyond the boundary, we clip the centroid coordinate such that the patch lies entirely inside the image). Then, we compute the maximum PSNR between the extracted patch and all valid 50×50 patches in the second frame whose centers are within a Chebyshev distance of at most 20 pixels from the first frame's centroid coordinate. To compute the PCons of an entire video, we take the average PCons over all consecutive frame pairs (*i.e.*, a 2-frame sliding window).

3. Additional Quantitative Results

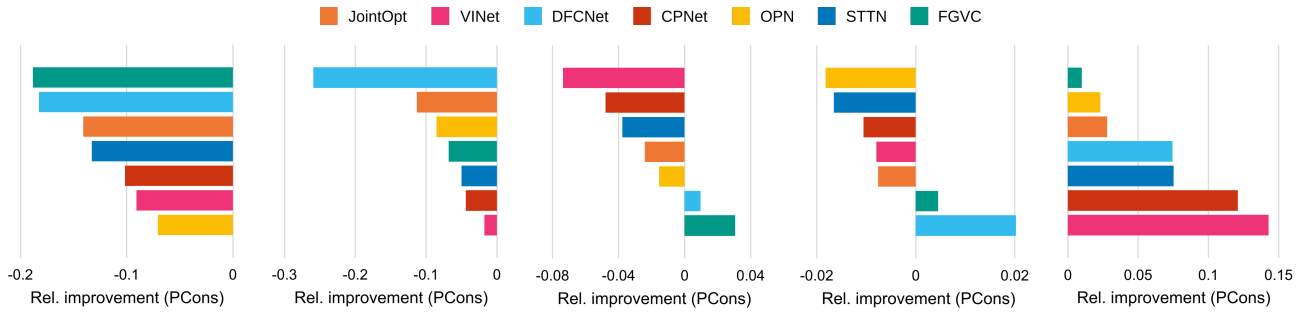
In Figure 1, we show the average performance of the seven evaluated inpainting methods on each of our ten DEVIL splits. We observe that across the reconstruction and realism metrics (Figure 1a-d), performance changes substantially under the occlusion mask attributes, but less substantially under source video attributes. The temporal consistency metric PCons changes less dramatically under the DEVIL attributes (Figure 1e), suggesting that temporal consistency performance is relatively stable under changes in the source video and mask content.

In Figure 2, we show the relative improvement experienced by each method when FG displacement, pose motion, and size are increased. The flow propagation methods FGVC, JointOpt, and DFCNet generally benefit the most from increased FG displacement and pose motion, whereas OPN benefits the least. As for FG size, OPN is more sensitive to this attribute than the other methods under three out of four reconstruction and realism performance metrics.

Figure 3 shows the relative change in temporal consistency performance (PCons) when each DEVIL attribute changes from low to high. Overall, we found that temporal consistency is the aspect of inpainting quality that is least sensitive to changes in DEVIL attributes; however, some models still experience more noticeable differences than others (*e.g.*, DFCNet is remarkably sensitive to BG scene motion).

References

[1] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. SURF: Speeded Up Robust Features. In Aleš Leonardis, Horst Bischof, and Axel Pinz, editors, *European Conference on Computer Vision*, Lecture Notes in Computer Science, pages 404–417, Berlin, Heidelberg,



(a) Camera motion (b) BG scene motion (c) FG displacement (d) FG pose motion (e) FG size

Figure 3: Relative improvement in temporal consistency when DEVIL attributes change from low to high.

2006. Springer. 1

[2] Joao Carreira and Andrew Zisserman. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. pages 6299–6308, 2017. 2

[3] Ya-Liang Chang, Zhe Yu Liu, Kuan-Ying Lee, and Winston Hsu. Free-Form Video Inpainting With 3D Gated Convolution and Temporal PatchGAN. In *IEEE International Conference on Computer Vision*, Oct. 2019. 1

[4] Martin A. Fischler and Robert C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*, 24(6):381–395, June 1981. 1

[5] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask R-CNN. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2961–2969, 2017. 1

[6] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In *Neural Information Processing Systems*, NIPS'17, pages 6629–6640, Long Beach, California, USA, Dec. 2017. Curran Associates Inc. 2

[7] SmugMug Inc. Flickr. https://www.flickr.com/. 1

[8] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *Neural Information Processing Systems*, 2012. 2

[9] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common Objects in Context. In *European Conference on Computer Vision*, Lecture Notes in Computer Science, pages 740–755, Cham, 2014. Springer International Publishing. 1

[10] 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 *IEEE Conference on Computer Vision and Pattern Recognition*, pages 724–732, 2016. 1

[11] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going Deeper With Convolutions. pages 1–9, 2015. 3

[12] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2, 2019. https://github.com/facebookresearch/detectron2. 1

[13] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018. 2