Image Inpainting with External-internal Learning and Monochromic Bottleneck
Supplementary Material

1. Implementation Details

After experimenting with different network architectures (U-net, ResNet, SkipNet) for our progressive internal colorization network, we adopted a ResNet-like architecture since it is the fastest among all the models that generate high-quality results. The number of feature channels is set to 32, and the filter size is 3. The network is composed of 9 two-layer residual blocks, 1 input Conv layer, and 1 output Conv layer. The input channel number of the first generator is 1 (gray only), and for other generators is 4 (gray+RGB). The output channel for each generator is 3 (RGB). Note that all our experiments are conducted in the RGB color-space instead of Lab color-space, so the luminance of the input image can possibly change slightly. BatchNorm, reflection padding, and LeakyReLU are adopted.

We choose the pyramid height for our progressive color propagation based on the image size and image content. The default pyramid height is set to 3 for Places2 images. The corresponding iteration number and learning rate at each level is [500,1000,1000] and [0.01, 0.005, 0.003]. We empirically found that with a larger learning rate at a lower level, the output contains more diverse colors, and with a lower learning rate at a higher level, the output contains more fine-grained details. In case the default configuration dose not yield a compelling colorization results, we can tune the pyramid height and learning rates for better performance.

2. Feasibility of Internal Colorization

We also provide more examples to validate the feasibility of the proposed internal propagation. These examples are from different categories, including natural scenery, buildings, human faces, and animals. As shown in Fig. 1 and Fig. 2, our scheme achieves stable colorization results on various images with different mask ratios.

3. User-guided Inpainting

Users can control the color of inpainted content with our inpainting method. We provide more details of user-guided inpainting and comparison results in Fig. 3. In this example, our model utilizes only one extra user-guided color point as a hint to generate realistic eyes with different colors. In contrast, other guided colorization methods fail to produce visually pleasing results and show obvious color bleeding artifacts in the eyes.

4. More Results

More visual results are shown in Fig. 4, Fig. 5.

References

[1] Eduardo SL Gastal and Manuel M Oliveira. Domain transform for edge-aware image and video processing. In *ACM SIGGRAPH 2011 papers*, pages 1–12. 2011.

[2] Anat Levin, Dani Lischinski, and Yair Weiss. Colorization using optimization. In *ACM SIGGRAPH*. 2004.

[3] Zili Yi, Qiang Tang, Shekoofeh Azizi, Daesik Jang, and Zhan Xu. Contextual residual aggregation for ultra high-resolution image inpainting. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7508–7517, 2020.

[4] Richard Zhang, Jun-Yan Zhu, Phillip Isola, Xinyang Geng, Angela S Lin, Tianhe Yu, and Alexei A Efros. Real-time user-guided image colorization with learned deep priors. *ACM Transactions on Graphics (TOG)*, 9(4), 2017.

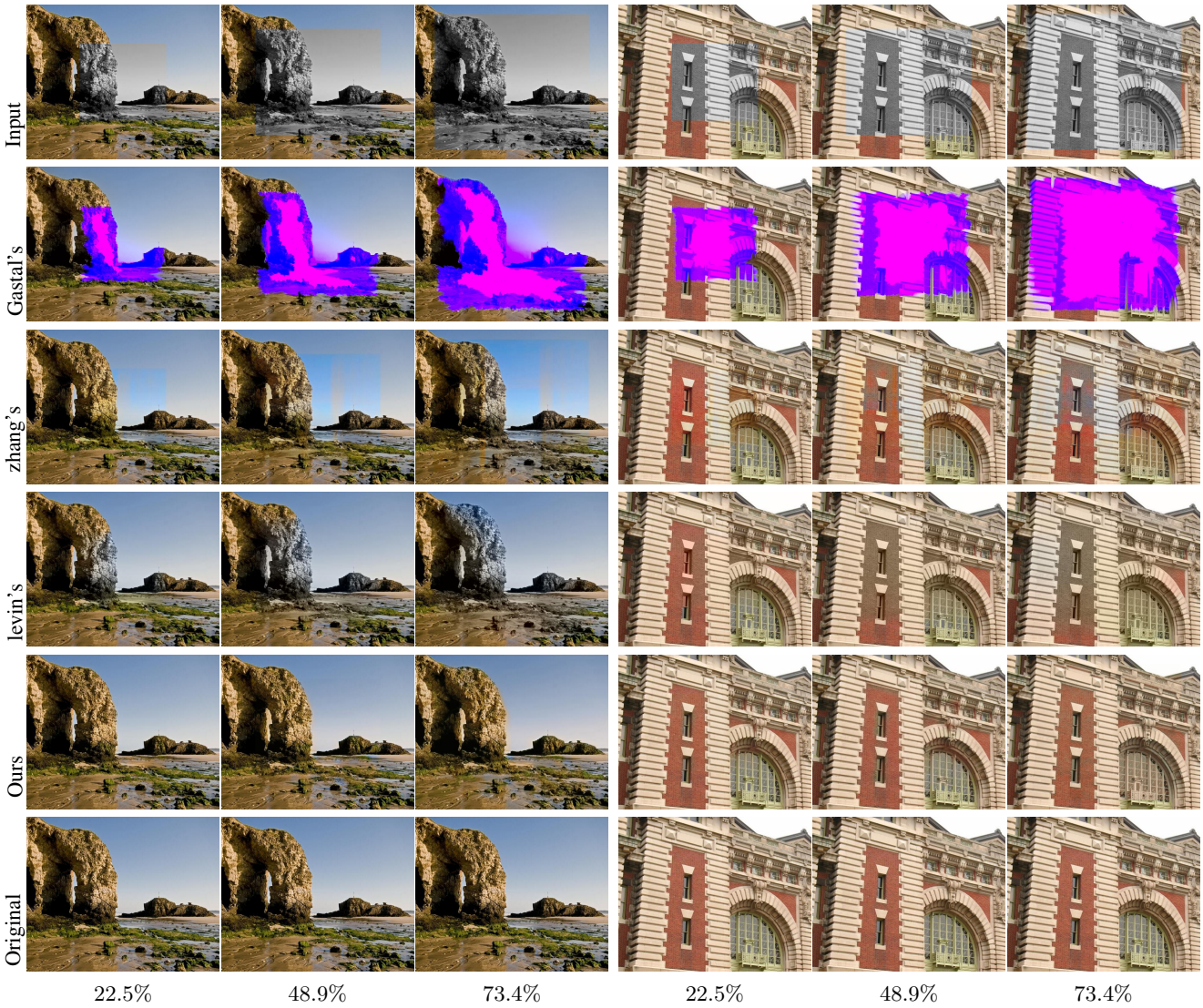

Figure 1: Feasibility of our internal colorization method. We increase the mask ratio of I_{hole} from 22.5% to 73.4% and colorize the ground-truth grayscale image with our internal colorization method. We also give colorization results by Zhang et al. [4], Gastal et al. [1] and Levin et al. [2] for comparison.

Figure 2: Feasibility of our internal colorization method. We increase the mask ratio of I_{hole} from 22.5% to 73.4% and colorize the ground-truth grayscale image with our internal colorization method. We also give colorization results by Zhang et al. [4], Gastal et al. [1] and Levin et al. [2] for comparison.

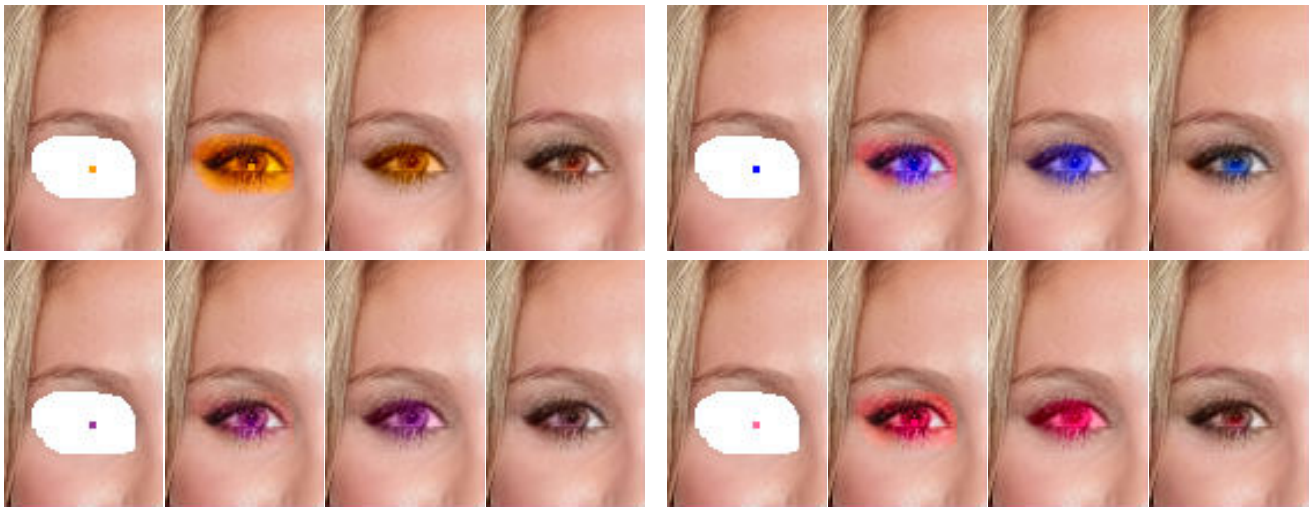

Figure 3: Examples of user-guided inpainting by our method. Users can control the color of inpainted content.

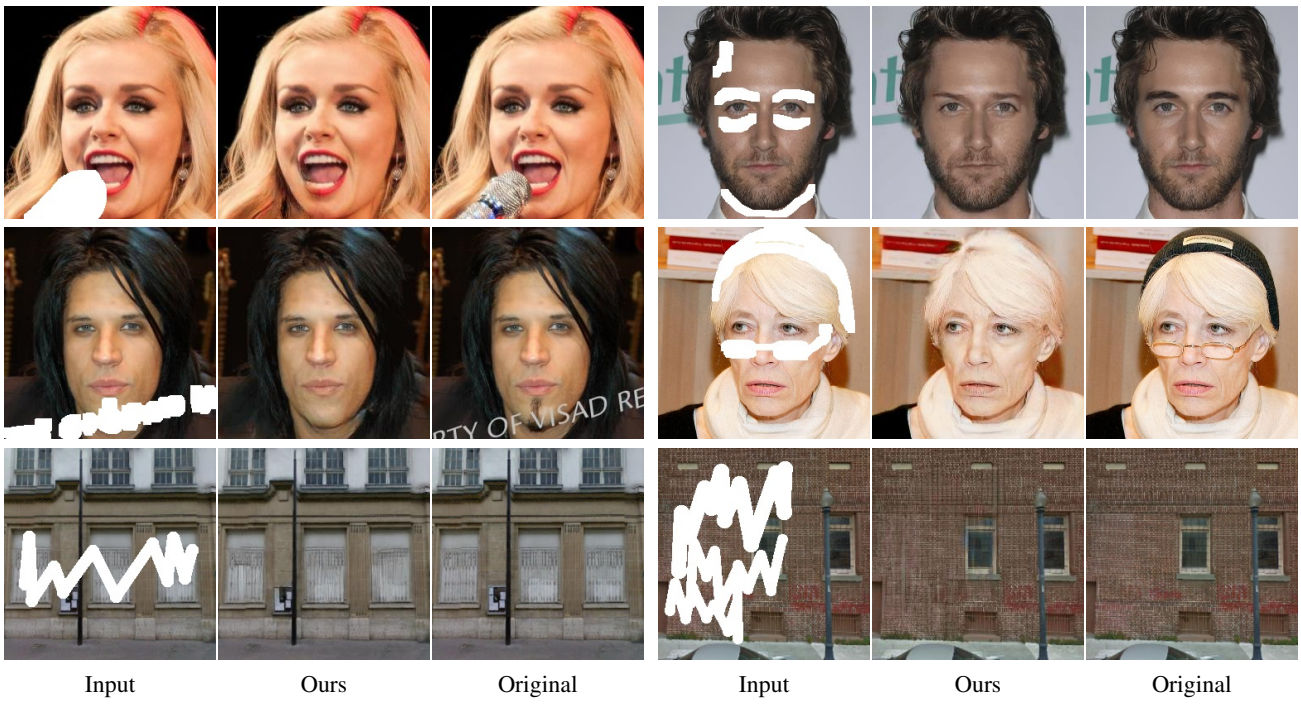

Figure 4: More results.

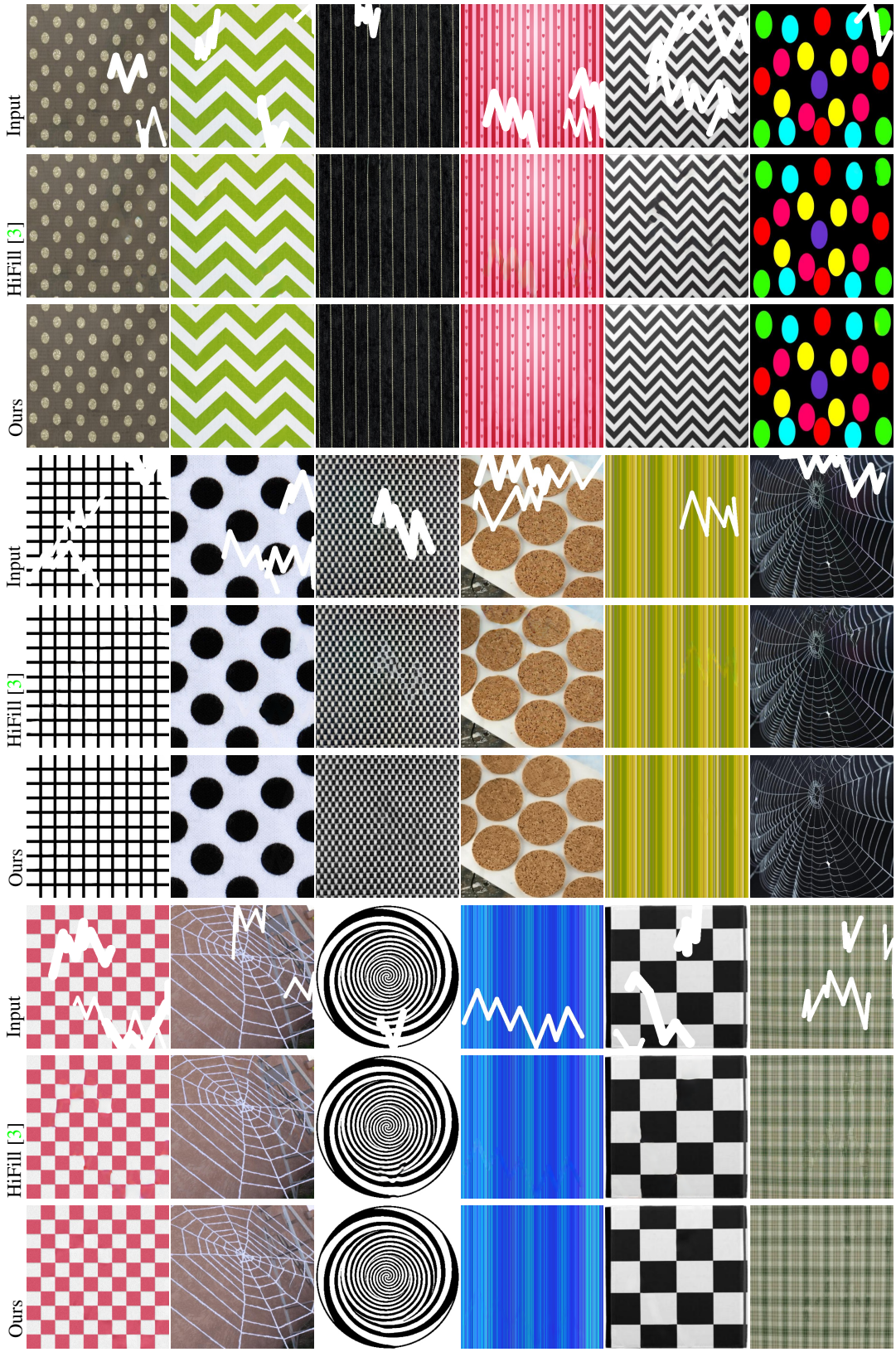

Figure 5: Sampled results on DTD.