Appendix

Shant Navasardyan[1[0000−0002−1999−9999]] and Marianna Ohanyan[2[0000−0002−4815−3802]]

[1] Picsart Inc., Armenia, Yerevan `shant.navasardyan@picsart.com`
[2] Picsart Inc., Armenia, Yerevan `marianna.ohanyan@picsart.com`

1 Details on Network Architecture

Here we describe the architectures of our networks in detail. Our generator network is composed of two parts, coarse and refinement. Let for the original image $I \in \mathbb{R}^{H\times W\times 3}$ and the binary mask $M \in \{0,1\}^{H\times W}$ the input of the coarse network be $I_m = (1 - M) \odot I$. The architecture of the coarse network can be found in the Table 1.

Let I_c be the output of the coarse model. Then we feed the refinement network by the image $I_{c\_comp} = (1-M) \odot I + M \odot I_c$. The architecture of the refinement network can be found in Table 2.

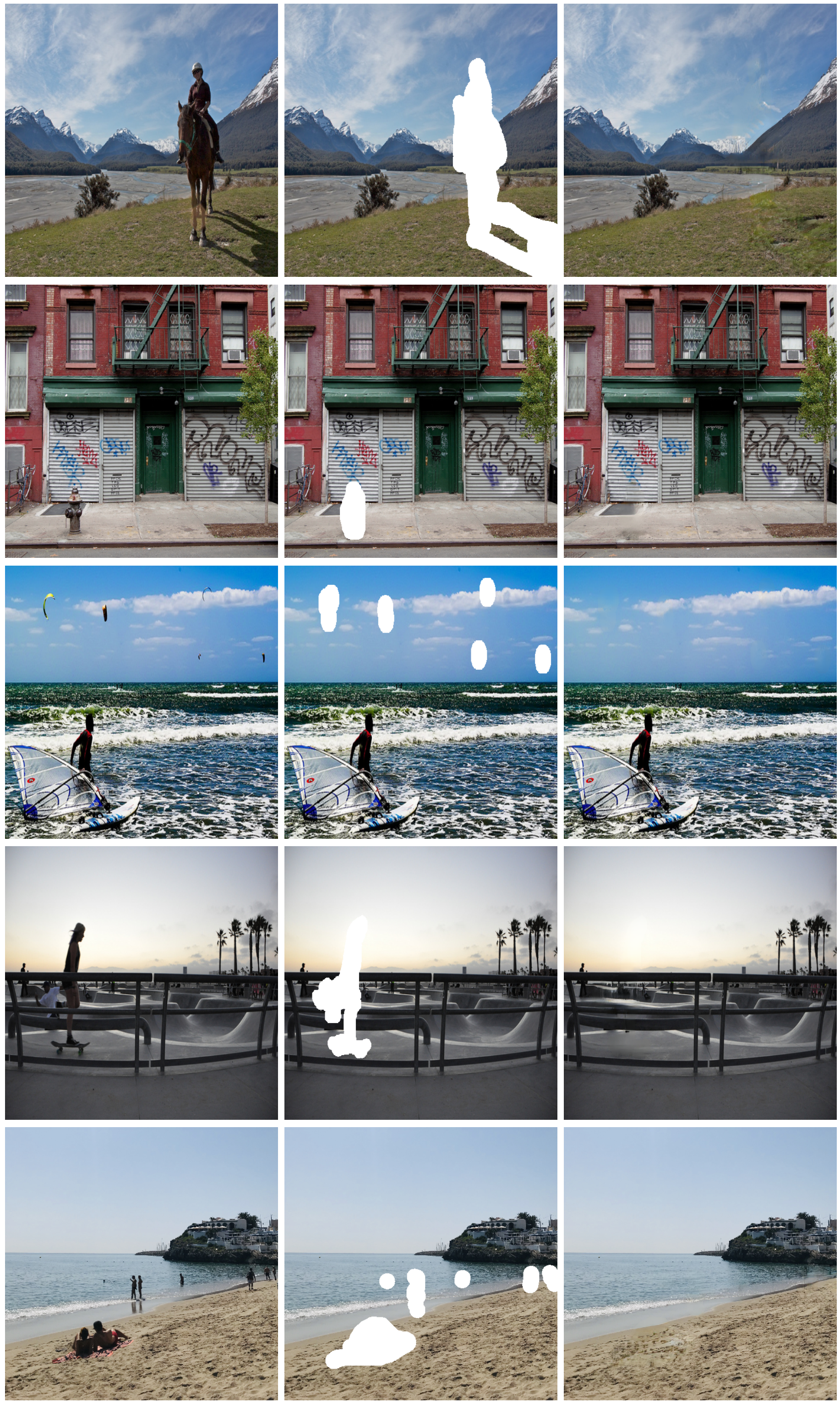
Let the output of the refinement network be I_r. Then we consider the image $I_{comp} = (1 - M) \odot I + M \odot I_r$ and pass it trough the discriminator. The architecture of the discriminator is presented in the Table 3.

2 More Visual Results

In this section we provide more results of our method and compare it with the methods [1, 4]. In Fig. 3 and Fig. 4 a part of the filled region is zoomed to show the difference between the methods. In Fig. 5 and Fig. 6 more images are shown for comparison. In Fig. 2 more images completed with our method are shown. In Fig. 1 high resolution (512×512) completed images are shown.

References

1. Liu, G., Reda, F.A., Shih, K.J., Wang, T.C., Tao, A., Catanzaro, B.: Image inpainting for irregular holes using partial convolutions. In: ECCV. (2018)
2. Clevert, D.A., Unterthiner, T., Hochreiter, S.: Fast and accurate deep network learning by exponential linear units (elus). CoRR **abs/1511.07289** (2015)
3. Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.S.: Generative image inpainting with contextual attention. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2018) 5505–5514
4. Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.: Free-form image inpainting with gated convolution. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV). (2019) 4470–4479



Original *Masked* *Our result on* 512×512

Fig. 1. Results of our method on high resultion.

Original *Masked* *Our result on* 256×256

Fig. 2. Results of our method.

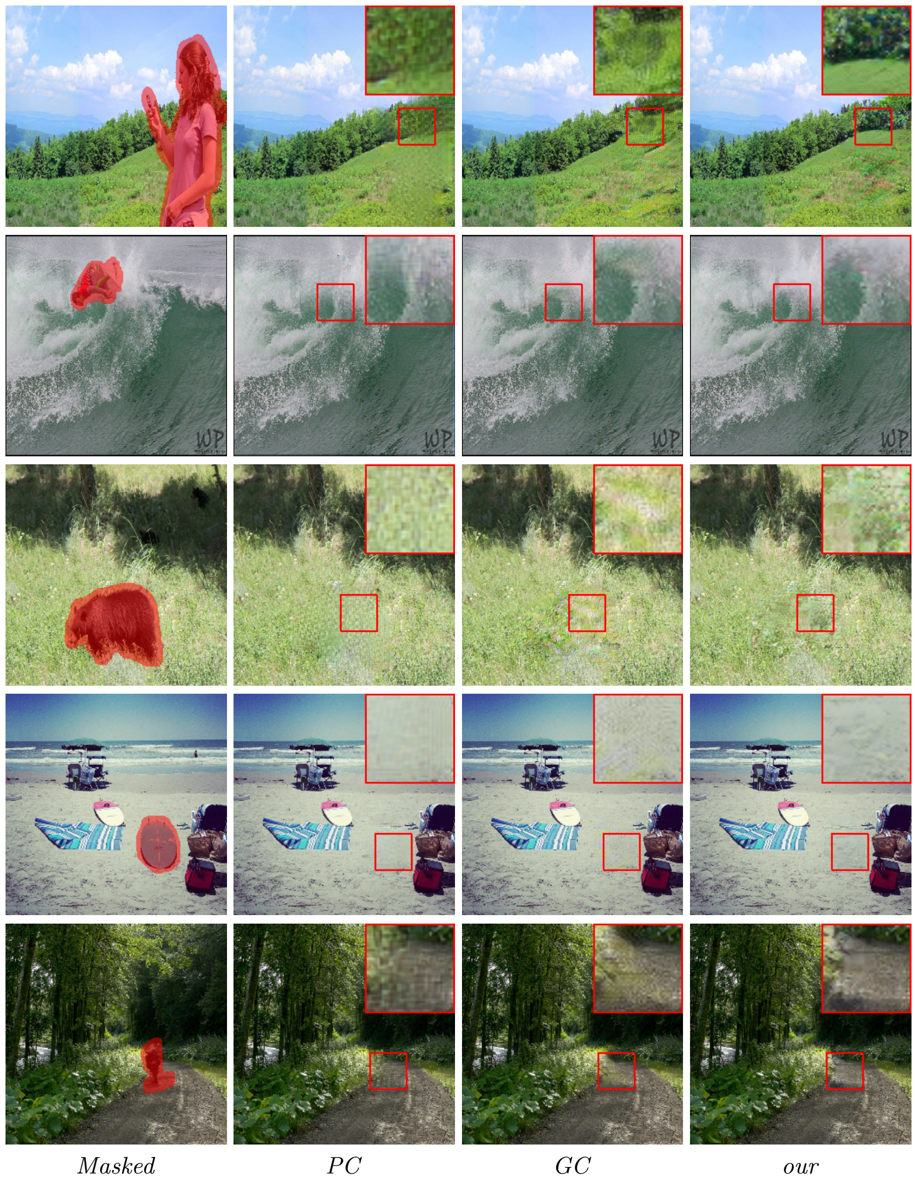


Fig. 3. Zoomed comparisons of our method with PC [1] and GC [4].

Fig. 4. Zoomed comparisons of our method with PC [1] and GC [4].

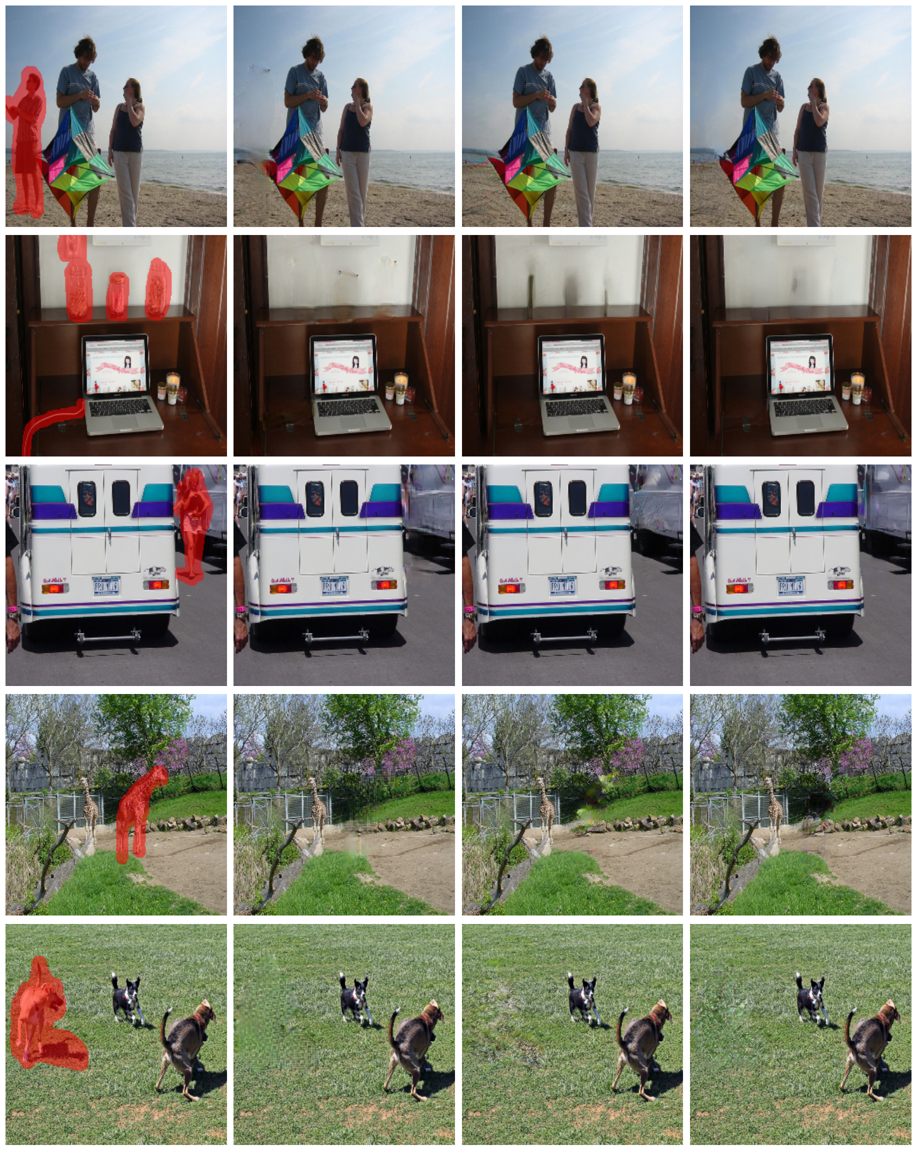


Fig. 5. Comparisons of our method with PC [1] and GC [4].

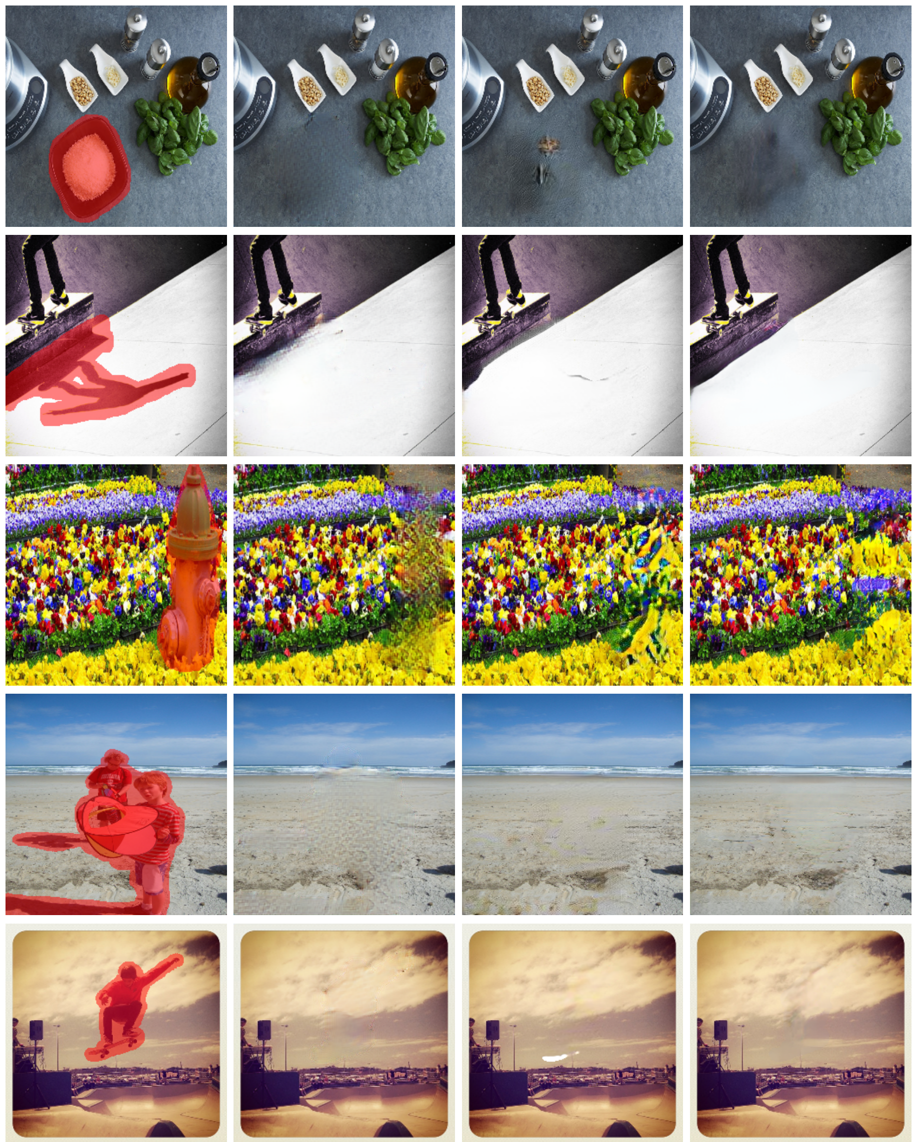


Fig. 6. Comparisons of our method with PC [1] and GC [4].

Table 1. The architecture of our coarse network. PConv_x denotes partial convolution layers [1]. *ELU* denotes the *Exponential Linear Unit* [2]. Conv_x is a convolution block composed of a convolution and a non-linearity. ConvUp_x is denoted a block composed of a nearest neighbor upsampling followed by a convolution and a non-linearity. All paddings are of the type "same". The parameter $T = \infty$ means that we iteratively fill missing region peels until the missing region is filled completely.

Layer Name	Parameters
PConv_1	$filters = 48, ksize = 5, strides = 1, activation = ELU$
PConv_2	$filters = 96, ksize = 3, strides = 2, activation = ELU$
PConv_3	$filters = 96, ksize = 3, strides = 1, activation = ELU$
PConv_4	$filters = 192, ksize = 3, strides = 2, activation = ELU$
PConv_5	$filters = 192, ksize = 3, strides = 1, activation = ELU$
PConv_6	$filters = 192, ksize = 3, strides = 1, activation = ELU$
Onion_Conv	$d = 8, k_f = 2, k_m = 4, T = \infty, filters = 96,$ $k_c = 3, strides = 1, activation = ELU$
Conv_7	$filters = 192, ksize = 3, strides = 2, activation = ELU$
Conv_8	$filters = 192, ksize = 3, strides = 1, activation = ELU$
Conv_9	$filters = 192, ksize = 3, strides = 1, activation = ELU$
Conv_10	$filters = 192, ksize = 3, strides = 2, activation = ELU$
Conv_11	$filters = 192, ksize = 3, strides = 1, activation = ELU$
Conv_12	$filters = 192, ksize = 3, strides = 1, activation = ELU$
ConvUp_13	$filters = 192, ksize = 3, activation = ELU$
Conv_14	$filters = 192, ksize = 3, strides = 1, activation = ELU$
ConvUp_15	$filters = 192, ksize = 3, activation = ELU$
Conv_16	$filters = 192, ksize = 3, strides = 1, activation = ELU$
ConvUp_17	$filters = 96, ksize = 3, activation = ELU$
Conv_18	$filters = 96, ksize = 3, strides = 1, activation = ELU$
ConvUp_19	$filters = 48, ksize = 3, activation = ELU$
Conv_20	$filters = 24, ksize = 3, strides = 1, activation = ELU$
Conv_21	$filters = 3, ksize = 3, strides = 1$
Tanh	

Table 2. The architecture of the refinement network. All paddings are of the type "same". *ReLU* denotes the *Rectified Linear Unit.* ContexAtt is the contextual attention layer [3].

Layer Name	Parameters
Conv_22	$filters = 48, ksize = 5, strides = 1, activation = ELU$
Conv_23	$filters = 48, ksize = 3, strides = 2, activation = ELU$
Conv_24	$filters = 96, ksize = 3, strides = 1, activation = ELU$
Conv_25	$filters = 96, ksize = 3, strides = 2, activation = ELU$
Conv_26	$filters = 192, ksize = 3, strides = 1, activation = ELU$
Conv_27	$filters = 192, ksize = 3, strides = 1, activation = ELU$
Conv_28	$filters = 192, ksize = 3, strides = 1, rate = 2, activation = ELU$
Conv_30	$filters = 192, ksize = 3, strides = 1, rate = 4, activation = ELU$
Conv_31	$filters = 192, ksize = 3, strides = 1, rate = 8, activation = ELU$
Conv_32	$filters = 192, ksize = 3, strides = 1, rate = 16, activation = ELU$
Conv_33 (on input)	$filters = 48, ksize = 5, strides = 1, activation = ELU$
Conv_34	$filters = 48, ksize = 3, strides = 2, activation = ELU$
Conv_35	$filters = 96, ksize = 3, strides = 1, activation = ELU$
Conv_36	$filters = 192, ksize = 3, strides = 2, activation = ELU$
Conv_37	$filters = 192, ksize = 3, strides = 1, activation = ELU$
Conv_38	$filters = 192, ksize = 3, strides = 1, activation = ReLU$
ContextAtt	$filters = 192, ksize = 3, strides = 1, rate = 2$
Conv_39	$filters = 192, ksize = 3, strides = 1, activation = ELU$
Conv_40	$filters = 192, ksize = 3, strides = 1, activation = ELU$
Concat (w Conv_32)	
Conv_41	$filters = 192, ksize = 3, strides = 1, activation = ELU$
Conv_42	$filters = 192, ksize = 3, strides = 1, activation = ELU$
ConvUp_40	$filters = 96, ksize = 3, activation = ELU$
Conv_41	$filters = 96, ksize = 3, strides = 1, activation = ELU$
ConvUp_42	$filters = 48, ksize = 3, activation = ELU$
Conv_43	$filters = 24, ksize = 3, strides = 1, activation = ELU$
Conv_44	$filters = 3, ksize = 3, strides = 1, activation = ELU$
Tanh	

Table 3. The architecture of the discriminator SNPatchGAN [4]. SConv_x denotes the convolution layer with a spectral normalization and a non-linearity. *LReLU* denotes the *Leaky Rectified Linear Unit* with a slope 0.2.

Layer Name	Parameters
SConv_1	$filters = 64, ksize = 5, strides = 2, activation = LReLU$
SConv_2	$filters = 128, ksize = 5, strides = 2, activation = LReLU$
SConv_3	$filters = 256, ksize = 5, strides = 2, activation = LReLU$
SConv_4	$filters = 256, ksize = 5, strides = 2, activation = LReLU$
SConv_5	$filters = 256, ksize = 5, strides = 2, activation = LReLU$
SConv_6	$filters = 256, ksize = 5, strides = 2, activation = LReLU$