Guide Your Eyes: Learning Image Manipulation under Saliency Guidance *Supplementary Materials*

Yen-Chung Chen*[1]
yenc.cs06g@nctu.edu.tw
Keng-Jui Chang*[1]
adplz53.cs06g@nctu.edu.tw
Yu-Chiang Frank Wang[3]
ycwang@ntu.edu.tw
Yi-Hsuan Tsai[2]
ytsai@nec-labs.com
Wei-Chen Chiu[1]
walon@cs.nctu.edu.tw

[1] National Chiao Tung University
[2] NEC Laboratories America
[3] National Taiwan University

1 Architecture

Here we provide more implementation details for our proposed method, *Saliency-Guidance Image Manipulation* (**SaGIM**).

1.1 Manipulation Network G

The network architecture of our manipulation network G (as shown in Table 1) is similar to the one used in the saliency estimation network E (i.e., SalGAN [2]) with several modifications. First, we switch the activation function from ReLU [1] to Leaky ReLU [4] in order to avoid the dying ReLU problem, which is critical especially when the gradients from the vital reconstruction loss $\mathcal{L}_{rec}$ in our model need to propagate a long way through E to reach each layer of G. Second, We slightly modify the first convolutional layer of G such that it supports the input with 4 channels, which is produced by the concatenation of the RGB input image and guiding saliency map. Last, as the network architecture G follows a convolutional autoencoder framework, the downsampling and upsampling operations in the encoder and decoder would cause loss of details in the image reconstruction. Toward eliminating this problem, we follow the idea proposed by [3] which is widely used in image segmentation or image-to-image translation tasks, to add skip connections (symmetric with respect to the bottleneck of autoencoder) across layers between the encoder and the decoder. Having skip connections in our architecture not only improves the quality of generated images, but also resolves the vanishing gradient problem which often appears in deep structures.

layer	channel	kernel	stride	activation
conv1	64	3×3	1	LeakyReLU
conv2	64	3×3	1	LeakyReLU
pool1		2×2	2	
conv3	128	3×3	1	LeakyReLU
conv4	128	3×3	1	LeakyReLU
pool2		2×2	2	
conv5	256	3×3	1	LeakyReLU
conv6	256	3×3	1	LeakyReLU
conv7	256	3×3	1	LeakyReLU
pool3		2×2	2	
conv8	512	3×3	1	LeakyReLU
conv9	512	3×3	1	LeakyReLU
conv10	512	3×3	1	LeakyReLU
pool4		2×2	2	
conv11	512	3×3	1	LeakyReLU
conv12	512	3×3	1	LeakyReLU
conv13	512	3×3	1	LeakyReLU
deconv1	512	3×3	1	LeakyReLU
deconv2	512	3×3	1	LeakyReLU
deconv3	512	3×3	1	LeakyReLU
upsample1		2×2	2	
deconv4	512	3×3	1	LeakyReLU
deconv5	512	3×3	1	LeakyReLU
deconv6	256	3×3	1	LeakyReLU
upsample2		2×2	2	
deconv7	256	3×3	1	LeakyReLU
deconv8	256	3×3	1	LeakyReLU
deconv9	128	3×3	1	LeakyReLU
upsample3		2×2	2	
deconv10	128	3×3	1	LeakyReLU
deconv11	64	3×3	1	LeakyReLU
upsample4		2×2	2	
deconv12	64	3×3	1	LeakyReLU
deconv13	64	3×3	1	LeakyReLU
deconv14	3	3×3	1	Sigmoid

Table 1: Architecture of manipulation network G.

1.2 Discriminator D

The architecture of discriminator D used in our SaGIM model is provided in Table 2.

layer	channel	kernel	stride	activation
conv1	32	3×3	1	LeakyReLU
pool1		2×2	2	
conv2	64	3×3	1	LeakyReLU
conv3	64	3×3	1	LeakyReLU
pool2		2×2	2	
conv4	64	3×3	1	LeakyReLU
conv5	64	3×3	1	LeakyReLU
pool3		2×2	2	
fc1	100	-	-	Tanh
fc2	2	-	-	Tanh
fc3	1	-	-	Sigmoid

Table 2: Architecture of discriminator D.

2 Ablation Study

We provide examples in Figure 1 for particularly showing the importance of cycle consistency which is uniquely introduced in our model. For instance, the results obtained by the model that is trained without cycle consistency could have unrealistic details and color shift.

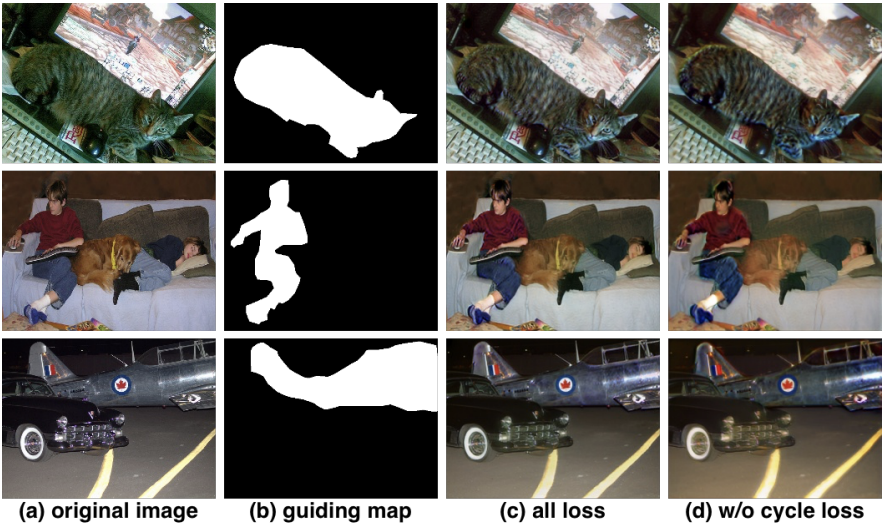

Figure 1: Example visualization for ablation study. Column **(c)** shows the results of having all losses activated, while column **(d)** shows the ones without cycle consistency loss.

3 Additional Results

In this section, we show more qualitative results for both manipulation tasks guided by the saliency map or the memorability measurement. In addition, we present an overview of the

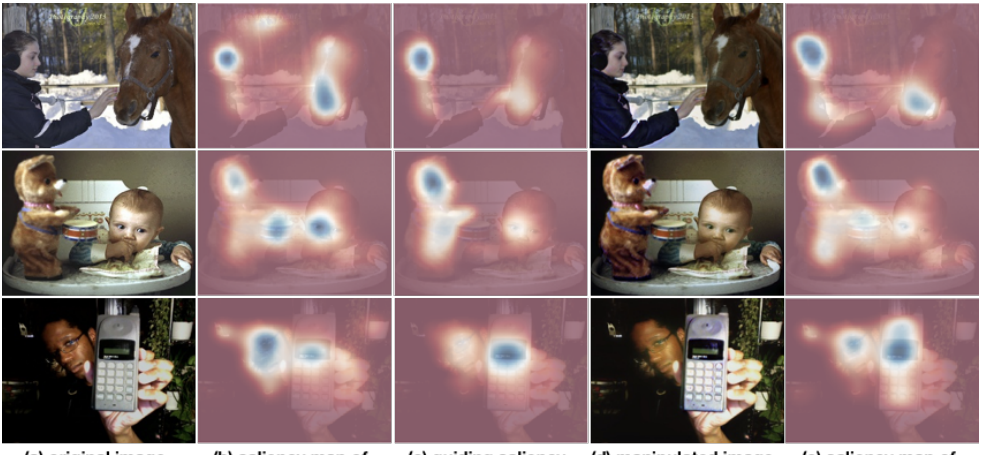

Figure 2: Example for saliency-guided image manipulation (better viewed in color).

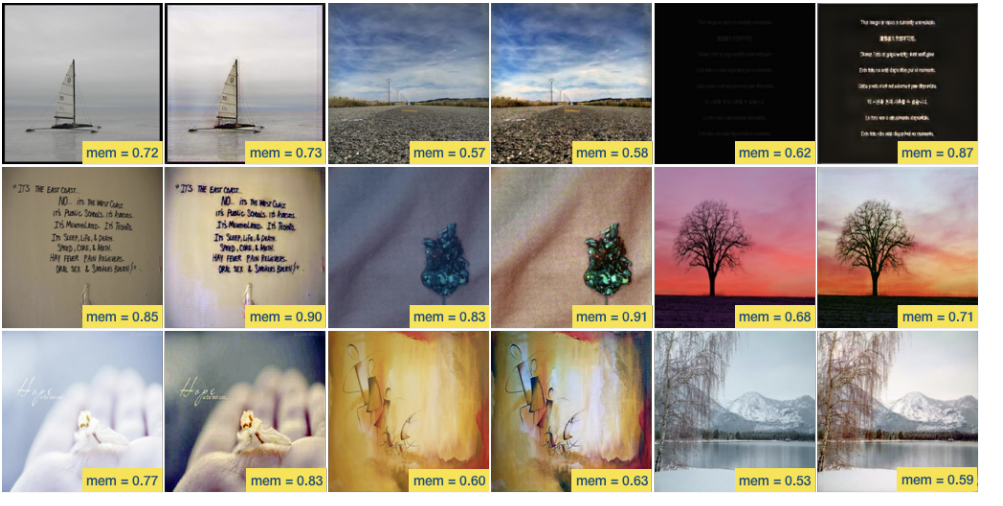

Figure 3: Example results of memorability-guided image manipulation based on our proposed pipeline. The left one in each pair is the original image, and the right one is the manipulation image. The corresponding memorability value for each image is recorded in the yellow box.

proposed method in the attached supplementary video.

3.1 Saliency-guided Image Manipulation

We provide additional example results for our saliency-guided image manipulation approach, as shown in Figure 2. Note that, here we visualize the saliency map (which is a single channel) by stacking it upon color images in order to provide their spatial correspondence.

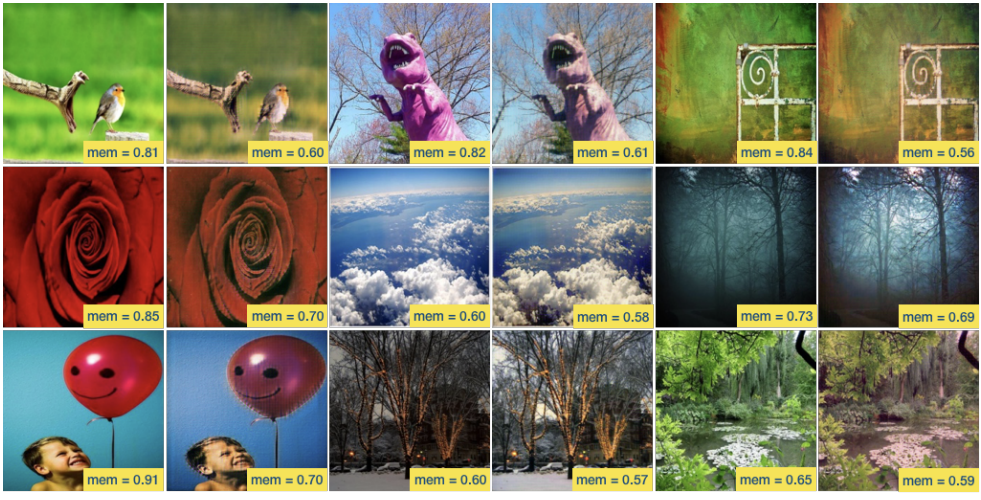

Figure 4: Example results of memorability-guided image manipulation based on our proposed pipeline. The left one in each pair is the original image, and the right one is the manipulation image. The corresponding memorability value for each image is recorded in the yellow box.

Overall, we observe that our method is able to manipulate original images based on the guiding saliency and in turn produces a saliency map that is consistent with the guiding one.

3.2 Memorability-guided Image Manipulation

In terms of memorability, we use random numbers as the guiding memorability in our pipeline. We categorize the results into two groups: 1) the case of ascending memorability, which collects those results whose memorability is greater than its corresponding original image, as shown in Figure 3, and 2) the case of descending memorability in Figure 4, in which the memorability scores of the images here are all lower than their original ones.

References

[1] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2010.

[2] Junting Pan, Elisa Sayrol, Xavier Giro-i Nieto, Cristian Canton Ferrer, Jordi Torres, Kevin McGuinness, and Noel E OâĂŹConnor. Salgan: Visual saliency prediction with adversarial networks. In *CVPR Scene Understanding Workshop*, 2017.

[3] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2015.

[4] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network. In *ICML Deep Learning Workshop*, 2015.