Supplementary Text: On Hallucinating Context and Background Pixels from a Face Mask using Multi-scale GANs

Sandipan Banerjee*[1], Walter J. Scheirer[2], Kevin W. Bowyer[2], and Patrick J. Flynn[2]
[1] Affectiva, USA
[2] Department of Computer Science & Engineering, University of Notre Dame, USA
sandipan.banerjee@affectiva.com, {wscheire, kwb, flynn}@nd.edu

Table 1: block_8 architecture (input size is 8×8×3)

Layer	Filter/Stride/Dilation	# of filters
conv0	3×3/1/2	128
conv1	3×3/1/2	1,024
RB1	3×3/1/1	1,024
fc1	512	-
fc2	16,384	-
conv2	3×3/1/1	4*512
PS1	-	-
conv3	5×5/1/1	3

Table 2: block_16 architecture (input size is 16×16×3)

Layer	Filter/Stride/Dilation	# of filters
conv0	3×3/1/2	128
conv1	3×3/2/1	512
RB1	3×3/1/1	512
conv2	3×3/2/1	1,024
RB2	3×3/1/1	1,024
fc1	512	-
fc2	16,384	-
conv3	3×3/1/1	4*512
PS1	-	-
conv4	3×3/1/1	4*256
PS2	-	-
conv5	5×5/1/1	3

Table 3: block_32 architecture (input size is 32×32×3)

Layer	Filter/Stride/Dilation	# of filters
conv0	3×3/1/2	128
conv1	3×3/2/1	256
RB1	3×3/1/1	256
conv2	3×3/2/1	512
RB2	3×3/1/1	512
conv3	3×3/2/1	1,024
RB3	3×3/1/1	1,024
fc1	512	-
fc2	16,384	-
conv3	3×3/1/1	4*512
PS1	-	-
conv4	3×3/1/1	4*256
PS2	-	-
conv5	3×3/1/1	4*128
PS3	-	-
conv6	5×5/1/1	3

1. Detailed Model Architecture

In this section, we list the layers of each generator block of our model. For both the cascaded and progressively growing (ProGAN) [8] versions of our model, the architectures of the generator block remain the same. For the cascaded model however, we use a set of four pixel shuffling [15] blocks to upscale the hallucination of a block 2x before feeding it as input to the next generator block. The architecture of each upscaling pixel shuffling blocks remains the same. The detailed layers of 'block_8', 'block_16', 'block_32', 'block_64', and 'block_128' layers are listed in Tables 1, 2, 3, 4, and 5 respectively. The convolution layers, residual blocks and pixel shuffling layers are indicated as 'conv', 'RB', and 'PS' respectively in the tables. For each of these layers in the generator, we used leaky *ReLU* with slope of 0.1 as the activation, except for the last 'conv' layer where a *tanh* activation is used [13, 14].

2. Ablation Studies

In this section, we analyze the effect of each component of our loss function on the overall quality of context and background synthesis. We present a comprehensive comparison that includes both qualitative results and quantitative experiments, using face images from the LFW dataset [7].

For this experiment, we prepare four variations of our multi-scale cascaded GAN model, while keeping the network architecture intact. We replace l_1 loss with l_2 loss

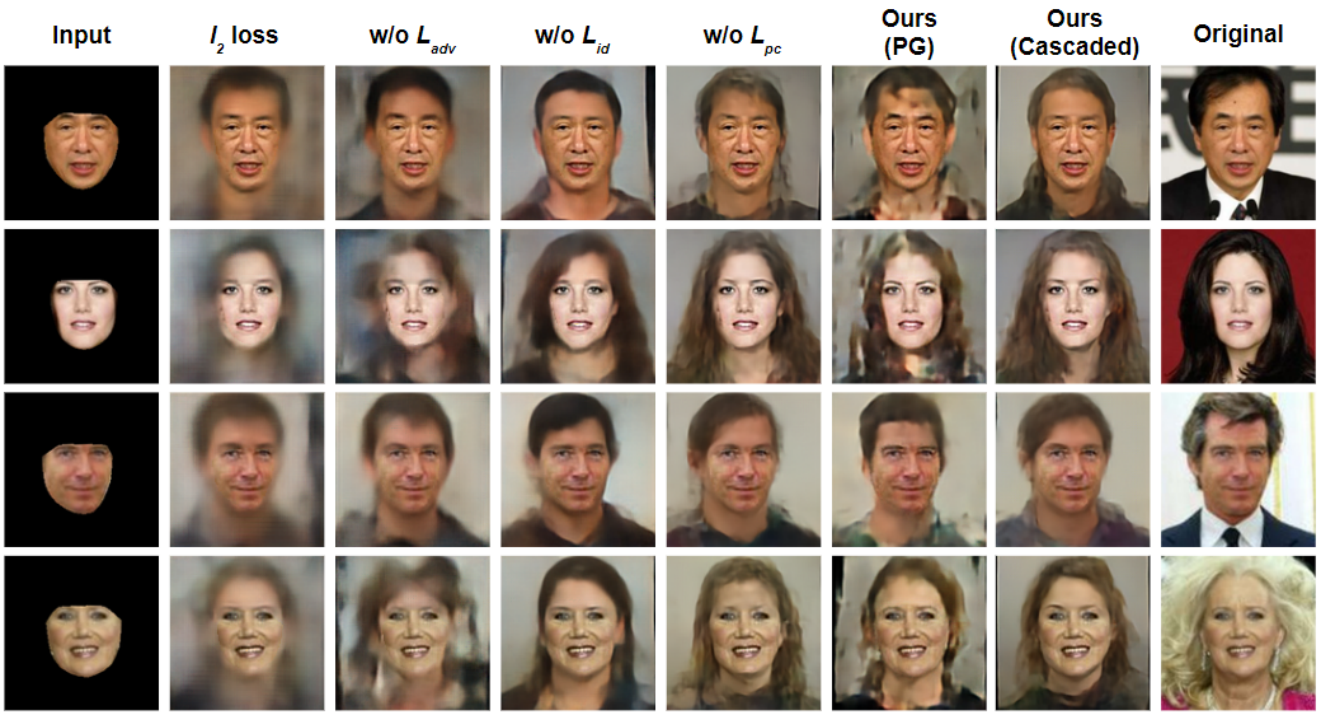

Figure 1: Ablation studies - hallucination results of our multi-scale GAN model and its variants.

Table 4: block_64 architecture (input size is 64×64×3)

Layer	Filter/Stride/Dilation	# of filters
conv0	3×3/1/2	128
conv1	3×3/2/1	128
RB1	3×3/1/1	128
conv2	3×3/2/1	256
RB2	3×3/1/1	256
conv3	3×3/2/1	512
RB3	3×3/1/1	512
conv4	3×3/2/1	1,024
RB4	3×3/1/1	1,024
fc1	512	-
fc2	16,384	-
conv3	3×3/1/1	4*512
PS1	-	-
conv4	3×3/1/1	4*256
PS2	-	-
conv5	3×3/1/1	4*128
PS3	-	-
conv6	3×3/1/1	4*64
PS4	-	-
conv7	5×5/1/1	3

Table 5: block_128 architecture (input size is 128×128×3)

Layer	Filter/Stride/Dilation	# of filters
conv0	3×3/1/2	128
conv1	3×3/2/1	64
RB1	3×3/1/1	64
conv2	3×3/2/1	128
RB2	3×3/1/1	128
conv3	3×3/2/1	256
RB3	3×3/1/1	256
conv4	3×3/2/1	512
RB4	3×3/1/1	512
conv5	3×3/2/1	1,024
RB5	3×3/1/1	1,024
fc1	512	-
fc2	16,384	-
conv3	3×3/1/1	4*512
PS1	-	-
conv4	3×3/1/1	4*256
PS2	-	-
conv5	3×3/1/1	4*128
PS3	-	-
conv6	3×3/1/1	4*64
PS4	-	-
conv7	3×3/1/1	4*64
PS5	-	-
conv8	5×5/1/1	3

as the metric for computing L_{pixel} for one model. For the other three models, we remove one of the other three losses (*i.e.*, L_{adv}, L_{id}, and L_{pc}) in each case. We keep the weight of the other loss components intact in each case. To analyze

Table 6: Ablation Studies - quantitative results on the LFW [7] dataset.

Model	Mean Match Score	Mean SSIM [16]	FID [6]	Mean Perceptual Error [12]
l_2 loss	0.520	0.413	166.76	2.489
w/o L_{adv}	0.522	0.411	132.71	2.320
w/o L_{id}	0.609	0.519	91.65	1.956
w/o L_{pc}	0.624	0.528	101.44	2.046
Ours (ProGAN)	0.668	0.466	103.71	2.255
Ours (Cascaded)	**0.722**	**0.753**	**46.12**	**1.256**

the role of the training regime, we compare each of these cascaded models with our ProGAN model keeping other factors constant. For this experiment, we use the same set of quality metrics as before - (1) mean match score with ResNet-50 [4], (2) mean SSIM [16], (3) FID [6], and (4) mean perceptual error [12] (description of each metric is available in Section 4 of main text). The quantitative results are presented in Table 6, along with visual results in Figure 1.

As expected, we find using l_2 loss for L_{pixel} drastically deteriorates the quality of the hallucinated face images by producing blurrier results. Since the pixel intensities are normalized to [0, 1], l_2 loss suppresses high frequency signals, compared to l_1, due to its squaring operation. The absence of a discriminator (w/o L_{adv}) at a network block fails to push the results towards the distribution of real face images, consequently hampering the performance of the model. Although not as critical as L_{pixel} and L_{adv}, the inclusion of both L_{id} and L_{pc} refine the hallucination result, as apparent from both the example images and the quality scores. The impact of the training regime, comparing end-to-end cascaded training with progressive growing (ProGAN), has already been discussed in Section 4 of the main text.

3. Epoch by Epoch Learning

To understand how the context and background are learned by the model during training, we save snapshots of our cascaded GAN model at different levels of training - 10 epochs, 20 epochs, 30 epochs, 40 epochs and 50 epochs. Except the training iterations, all other parameters and hyper-parameters remain the same. These models are then used to generate context and background pixels on masked face images from LFW [7]. Hallucinations for three such images have been shown in Figure 2.

As apparent from the figure, the model learns to generate a rough set of hair and skin pixels in the first few training epochs, not focusing on the clothes or background (10-20 epochs). Then it adds in pixels for the clothes and background, while further refining the overall skin and hair pixel quality (30-40 epochs). The validation loss stabilizes around the 50-th epoch (our hard termination point), and hence this snapshot has been used in our experiments. We also find the model to take a few extra iterations of refinement in hallucinating context and background for images with posed faces compared to those with frontal faces.

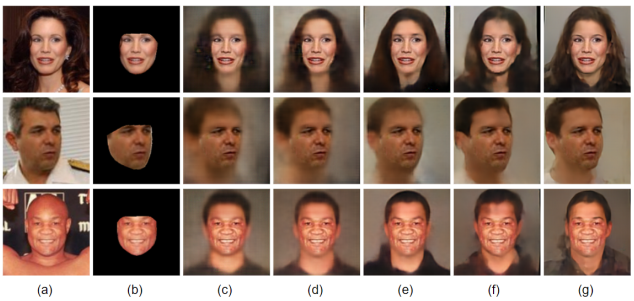

Figure 2: Sample synthesis results from LFW [7] at different levels of training - (a) the original face image (cropped), (b) masked face input, hallucination results after (c) 10 epochs, (d) 20 epochs, (e) 30 epochs, (f) 40 epochs, and (g) 50 epochs of training.

4. Changing the Background Pixels

To add more variety to our images, we add a post-processing step to further change the background pixels, while keeping the face and context pixels unchanged, using background images supplied by the user. We first locate the pixels outside the background (context + face mask) using the segmentation network from [21, 20, 18]. The pixels with the label 'Person' are kept inside the mask, which is further refined by a saliency map. This saliency map is computed using the gradient of each pixel of the image and the outer contour detected as the salient edge. The union of the initial mask and the points inside this contour produces the final foreground mask. Alternatively, the foreground mask can also be generated using the image matting network provided in [19]. The new background image is then blended in with the help of this foreground mask using a Laplacian pyramid based blending [2, 1].

5. Additional Qualitative Results

In this section, we present additional qualitative results for visual perusal. Face images, varying in gender, ethnicity, age, pose, lighting and expression, are randomly selected from the LFW dataset [7] and IJB-B [17] video frames. Each image is then aligned about their eye centers using landmark points extracted from Dlib [9], face masked

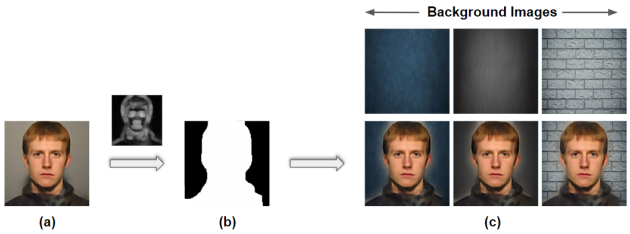

Figure 3: Background replacement process - (a) hallucinated face image (b) the detected foreground mask using a combination of gradient map and the segmentation network from [21, 20, 18], and (c) background pixels replaced with Laplacian blending [2].

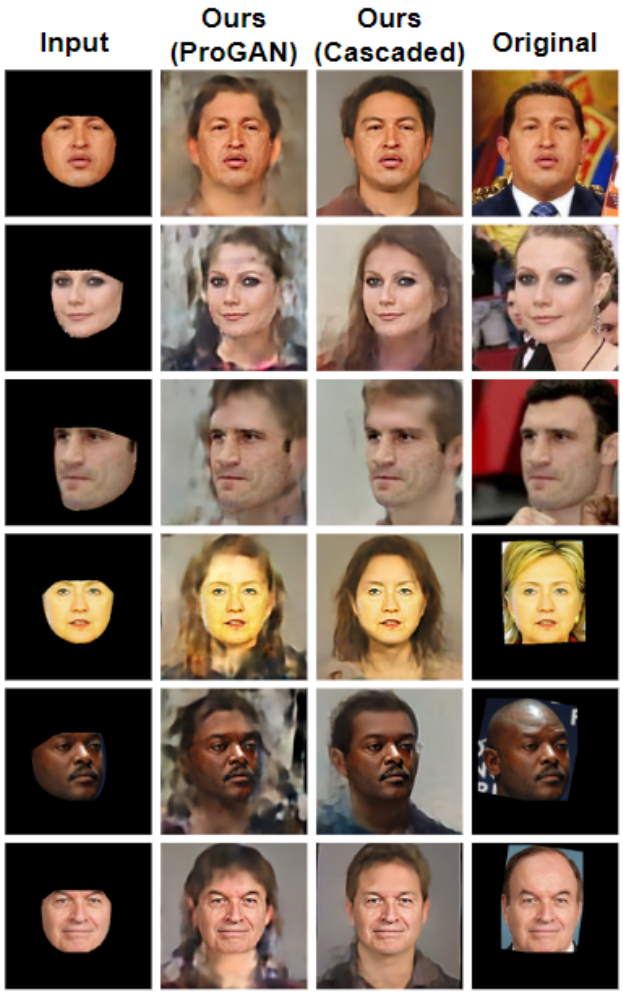

Figure 4: Additional qualitative results generated by our ProGAN and cascaded models. The first three rows are samples from the LFW [7] dataset, while the last three rows are taken from the IJB-B [17] dataset. All images are 128×128 in size.

and resized to 128×128. Each image is then fed to the trained snapshots, used in our original experiments, of our cascaded and progressively growing models for context and background pixel synthesis. The results are shown in Figure 4.

Figure 5: Some problematic cases - missing pixels for the microphone occluding subject's chin (left), no matching temples generated for the eyeglasses (middle), and hairstyle of wrong gender (right).

6. Model Limitations

As our model learns to hallucinate from the training data, we observe visual artifacts for face masks which vary drastically in appearance from it. For example, it fails to hallucinate missing pixels of occluding objects present in the face mask (like the microphone in leftmost image in Figure 5). This can be fixed by refining the input face mask to remove such occluding objects. In some cases our model mis-labels the gender of the face mask and generates the wrong hairstyle. Such an example can be seen Figure 5 (rightmost image), where the input male subject gets a female hairstyle. This issue can be resolved by either training two networks separately with male and female subjects or by adding a gender preserving loss (using [10]) to the loss function. Our model also fails to generate matching temples when the subject wears eyeglasses due to their absence in the training images (Figure 5 middle image). To tackle this issue, the training data can be augmented by adding eyeglasses to some images using [11, 5, 3].

References

[1] S. Banerjee, J. Bernhard, W. Scheirer, K. Bowyer, and P. Flynn. Srefi: Synthesis of realistic example face images. In *IJCB*, 2017. 3

[2] P. Burt and E. Adelson. The laplacian pyramid as a compact image code. In *IEEE Trans. on Communications*, volume 31, pages 532–540, 1983. 3, 4

[3] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Chool. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *CVPR*, 2018. 4

[4] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CVPR*, 2016. 3

[5] Z. He, W. Zuo, M. Kan, S. Shan, and X. Chen. Attgan: Facial attribute editing by only changing what you want. In *arXiv:1711.10678*, 2017. 4

[6] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*, 2017. 3

[7] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. In *Tech Report 07–49*, 2007. 1, 3, 4

[8] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *ICLR*, 2018. 1

[9] D. E. King. Dlib-ml: A machine learning toolkit. In *Journal of Machine Learning Research*, volume 10, pages 1755–1758, 2009. 3

[10] G. Levi and T. Hassner. Age and gender classification using convolutional neural networks. In *CVPR Workshops*, 2015. 4

[11] R. Natsume, T. Yatagawa, and S. Morishima. Rsgan: Face swapping and editing using face and hair representation in latent spaces. *arXiv:1804.03447*, 2018. 4

[12] E. Prashnani, H. Cai, Y. Mostofi, and P. Sen. Pieapp: Perceptual image-error assessment through pairwise preference. In *CVPR*, 2018. 3

[13] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR*, 2016. 1

[14] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. In *NIPS*, 2016. 1

[15] W. Shi, J. Caballero, F. Huszar, J. Totz, A. Aitken, R. Bishop, D. Rueckert, and Z. Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *CVPR*, 2016. 1

[16] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. on Image Processing*, 13(4):600–612, 2004. 3

[17] C. Whitelam, E. Taborsky, A. Blanton, B. Maze, J. Adams, T. Miller, N. Kalka, A. K. Jain, J. A. Duncan, K. Allen, J. Cheney, and P. Grother. Iarpa janus benchmark-b face dataset. In *CVPR Workshops*, 2017. 3, 4

[18] T. Xiao, Y. Liu, B. Zhou, Y. Jiang, and J. Sun. Unified perceptual parsing for scene understanding. *arXiv preprint*, 2018. 3, 4

[19] N. Xu, B. Price, S. Cohen, and T. Huang. Deep image matting. In *CVPR*, 2017. 3

[20] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba. Semantic understanding of scenes through the ade20k dataset. *arXiv:1608.05442*, 2016. 3, 4

[21] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba. Scene parsing through ade20k dataset. In *CVPR*, 2017. 3, 4