LEGAN: Disentangled Manipulation of Directional Lighting and Facial Expressions whilst Leveraging Human Perceptual Judgements (Supplementary Text)

Sandipan Banerjee[1], Ajjen Joshi[1], Prashant Mahajan*[2], Sneha Bhattacharya*[3], Survi Kyal[1] and Taniya Mishra*[4]

[1]Affectiva, USA, [2]Amazon, USA, [3]Silver Spoon Animation, USA, [4]SureStart, USA

{firstname.lastname}@affectiva.com, prmhp@amazon.com, snehabhattac@umass.edu, taniya.mishra@mysurestart.com

1. Quality Estimation Model: Architecture Details

We share details of the architecture of our quality estimator Q in Table 1. The fully connected layers in Q are denoted as 'fc' while each convolution layer, represented as 'conv', is followed by Leaky ReLU [20] activation with a slope of 0.01.

Table 1: Detailed architecture of our quality estimation model Q (input size is 128×128×3).

Layer	Filter/Stride/Dilation	# of filters
input	128×128	3
conv0	4×4/2/1	64
conv1	4×4/2/1	128
conv2	4×4/2/1	256
conv3	4×4/2/1	512
conv4	4×4/2/1	1024
conv5	4×4/2/1	2048
fc0	-	256
fc1	-	1

2. Quality Estimation Model: Naturalness Rating Distribution in Training

In this section, we share the distribution of the naturalness ratings that we collected from the Amazon Mechanical Turk (AMT) experiment (Stage II). To do this, we average the perceptual rating for each synthetic face image from its three scores and increment the count of a particular bin in [(0 - 1), (1 - 2), ... , (8 - 9), (9 - 10)] based on the mean score. As described in Section 3 of the main paper, we design the AMT task such that a mean rating between 0 and 5 suggests the synthetic image to look 'unnatural' while a score between 5 and 10 advocates for its naturalness. As can be seen in Figure 1, majority of the synthetic images used in our study generates a mean score that falls on the 'natural' side, validating their realism. When used to train our quality estimation model Q, these images tune its weights to look for the same perceptual features in other images while rating their naturalness.

*Work done while at Affectiva

Table 2: Hourglass architecture for expression mask (M_e) synthesis in the generator G. The input size is 128×128×9, three RGB channels (I_a) and six expression channels (c_e).

Layer	Filter/Stride/Dilation	# of filters
input	128×128/-/-	9
conv0	7×7/1/1	64
conv1	4×4/2/1	128
conv2	4×4/2/1	256
RB0	3×3/1/1	256
RB1	3×3/1/1	256
RB2	3×3/1/1	256
RB3	3×3/1/1	256
RB4	3×3/1/1	256
RB5	3×3/1/1	256
PS0	-	256
conv3	4×4/1/1	128
PS1	-	128
conv4	4×4/1/1	64
conv5 (M_e)	7×7/1/1	3

To further check the overall perceptual quality of each of the different synthesis approaches used in our study [10, 11, 1, 15, 3], we separately find the mean rating for each synthetic face image generated by that method, depicted in Figure 2. It comes as no surprise for the StyleGAN [11] images to rank the highest, with a mean score over 7, as its face images were pre-filtered for quality [2]. The other four approaches perform roughly the same, generating a mean score that falls between 6 and 7.

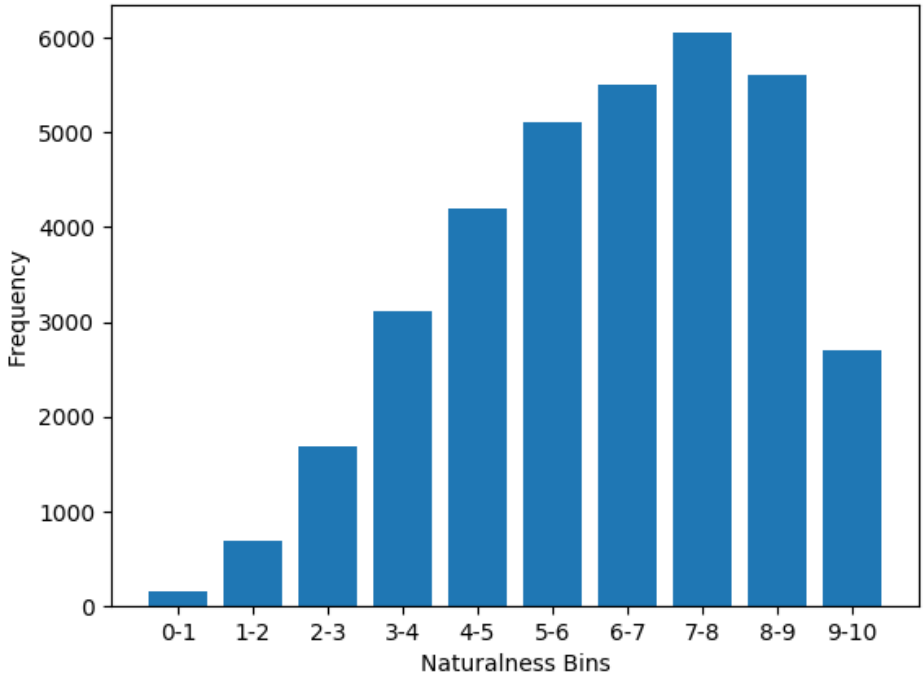

Figure 1: Histogram depicting the number of images in each naturalness bin, as rated by the Amazon Mechanical Turkers. Much more images fell on the 'natural' half (5 - 10) rather than the 'unnatural' one (0 - 5), suggesting the synthetic face images used in our study to be more or less realistic.

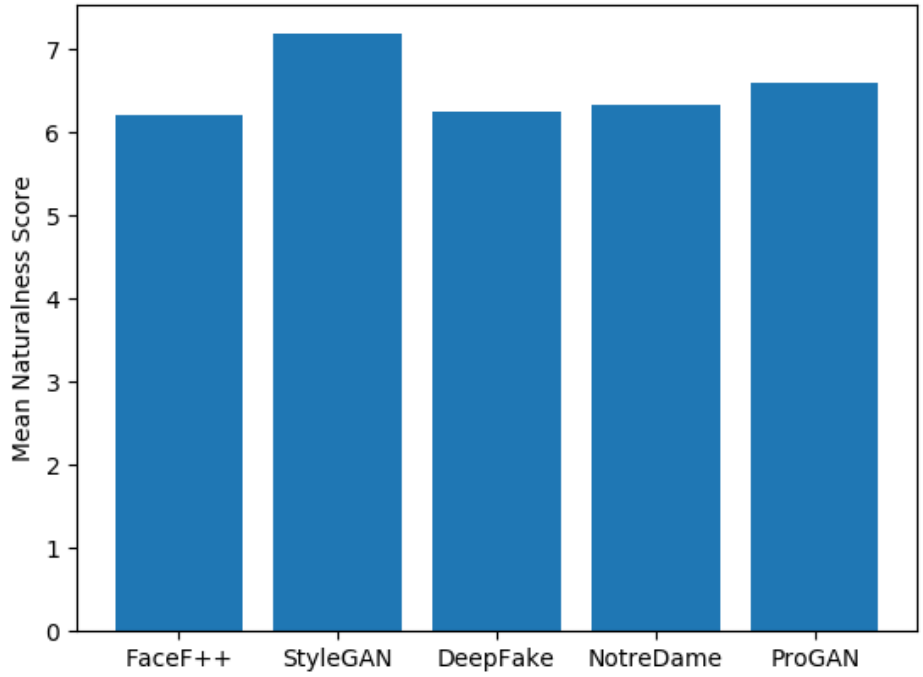

Figure 2: Mean naturalness rating of the different synthesis approaches used in our study [10, 11, 1, 15, 3]. As expected, the StyleGAN [11] images are rated higher than others as they were pre-filtered for quality[2].

3. Quality Estimation Model: Loss Function

Our loss uses the L2 norm between the predicted quality (p) and mean label (μ) and then computes a second L2 norm between this distance and the standard deviation (σ). σ acts as a margin in this case. If we consider μ as the center of a circle with radius of σ, then our loss tries to push p towards the boundary to fully capture the subjective-

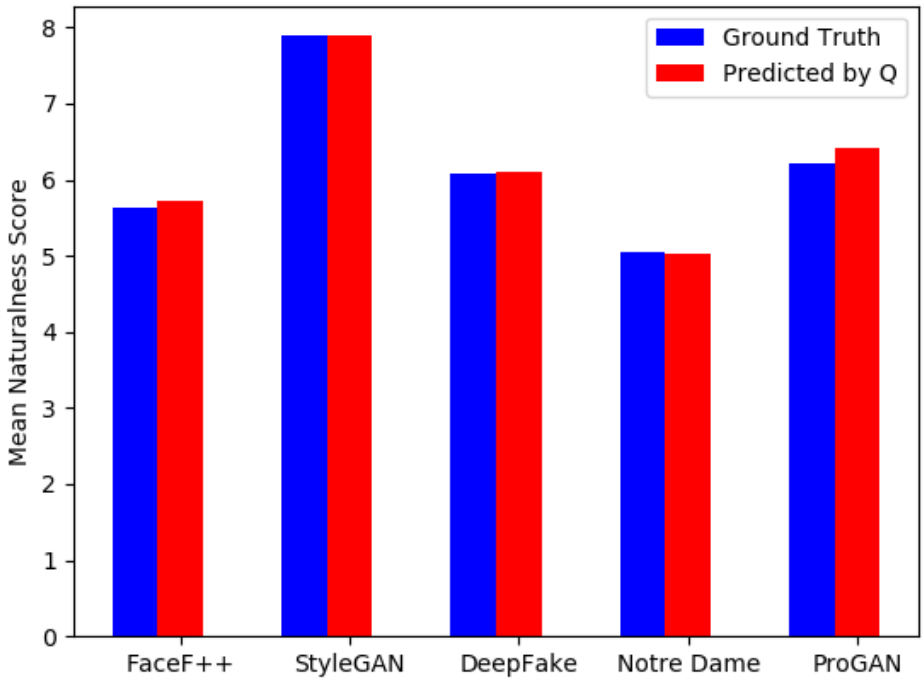

Figure 3: Mean naturalness rating, as estimated by Turkers (blue) and predicted by our trained quality estimation model Q (red), for the different synthesis approaches used in our study [10, 11, 1, 15, 3]. These ratings are specifically for images from the test split in our experiments, so Q never encountered them during training. Yet, Q is able to predict the naturalness of these images with a high degree of certainty.

Table 3: Hourglass architecture for lighting mask (M_l) synthesis in the generator G. The input size is 128×128×23, three RGB channels (I_a) and twenty expression channels (c_l).

Layer	Filter/Stride/Dilation	# of filters
input	128×128/-/-	23
conv0	7×7/1/1	64
conv1	4×4/2/1	128
conv2	4×4/2/1	256
RB0	3×3/1/1	256
RB1	3×3/1/1	256
RB2	3×3/1/1	256
RB3	3×3/1/1	256
RB4	3×3/1/1	256
RB5	3×3/1/1	256
PS0	-	256
conv3	4×4/1/1	128
PS1	-	128
conv4	4×4/1/1	64
conv5 (M_l)	7×7/1/1	3

Table 4: Hourglass architecture for target image ($G(I_a, f_b)$) synthesis in the generator G. The input size is 128×128×6, three expression mask channels (M_e) and three lighting mask channels (M_l).

Layer	Filter/Stride/Dilation	# of filters
input	128×128/-/-	6
conv0	7×7/1/1	64
conv1	4×4/2/1	128
conv2	4×4/2/1	256
RB0	3×3/1/1	256
RB1	3×3/1/1	256
RB2	3×3/1/1	256
RB3	3×3/1/1	256
RB4	3×3/1/1	256
RB5	3×3/1/1	256
PS0	-	256
conv3	4×4/1/1	128
PS1	-	128
conv4	4×4/1/1	64
conv5 ($G(I_a, f_b)$)	7×7/1/1	3

ness of human perception. We also tried a hinge version of this loss: $\max\left(0, \left(\|\mu - p\|_2^2 - \sigma\right)\right)$. This function penalizes p falling outside the permissible circle while allowing it to lie anywhere within it. When σ is low, both functions act similarly. We found the quality estimation model's (Q) predictions to be less stochastic when trained with the margin loss than the hinge. On a held-out test set, both losses performed similarly with only 0.2% difference in regression accuracy. Experimental results with LEGAN, and especially StarGAN trained using Q (Tables 1, 2, 3 in the main text), underpin the efficiency of the margin loss in comprehending naturalness. The improvements in perceptual quality, as demonstrated by LPIPS and FID, further justify its validity as a good objective for training Q.

Figure 4: Perceptual quality predictions by our trained quality estimation model (Q) on sample test images generated using [11, 10, 1, 15, 3]. For each image, the (mean $\pm$ standard deviation) of the three naturalness scores, collected from AMT, is shown below while Q's prediction is shown above in red.

Table 5: Detailed architecture of LEGAN's discriminator D (input size is 128×128×3).

Layer	Filter/Stride/Dilation	# of filters
input	128×128	3
conv0	4×4/2/1	64
conv1	4×4/2/1	128
conv2	4×4/2/1	256
conv3	4×4/2/1	512
conv4	4×4/2/1	1024
conv5	4×4/2/1	2048
conv6 (D_{src})	3×3/1/1	1
conv7 (D_{cls})	1×1/1/1	26

4. Quality Estimation Model: Prediction Accuracy During Testing

As discussed in Section 3 of the main text, we hold out 10% of the crowd-sourced data (3,727 face images) for testing our quality estimation model Q post training. Since Q never encountered these images during training, we use them to evaluate the effectiveness of our model. We separately compute the mean naturalness score for each synthesis approach used in our study and compare this value with the average quality score as predicted by Q. The results can be seen in Figure 3. Overall, our model predicts the naturalness score for each synthesis method with a high degree of certainty. Some qualitative results can also be seen in Figure 4.

5. LEGAN: Detailed Architecture

In this section, we list the different layers in the generator G and discriminator D of LEGAN. Since G is composed of three hourglass networks, we separately describe their architecture in Tables 2, 3 and 4 respectively. The convolution layers, residual blocks and pixel shuffling layers are indicated as 'conv', 'RB', and 'PS' respectively in the tables. After each of 'conv' and 'PS' layer in an hourglass, we use *ReLU* activation and instance normalization [18], except for the last 'conv' layer where a *tanh* activation is used [14, 16]. The description of D can be found in Table 5. Similar to Q, each convolution layer is followed by Leaky ReLU [20] activation with a slope of 0.01 in D, except for the final two

convolution layers that output the realness matrix D_{src} and the classification map D_{cls}.

6. LEGAN: Ablation Study

To analyze the contribution of each loss component on synthesis quality, we prepare 5 different versions of LEGAN by removing (feature disentanglement, L_{adv}, L_{cls}, L_{rec}, L_{qual}, and L_{id}) from G while keeping everything else the same. The qualitative and quantitative results, produced using MultiPIE [7] test data, are shown in Figure 5 and Table 6 respectively. For the quantitative results, the output image is compared with the corresponding target image in MultiPIE, and not the source image (*i.e.* input).

As expected, we find L_{adv} to be crucial for realistic hallucinations, in absence of which the model generates non-translated images totally outside the manifold of real images. The disentanglement of the lighting and expression via LEGAN's hourglass pair allows the model to independently generate transformation masks which in turn synthesize more realistic hallucinations. Without the disentanglement, the model synthesizes face images with pale-ish skin color and suppressed expressions. When L_{cls} is removed, LEGAN outputs the input image back as the target attributes are not checked by D anymore. Since the input image is returned back by the model, it generates a high face matching and mean quality score (Table 6, third row). When the reconstruction error L_{rec} is plugged off the output images lie somewhere in the middle, between the input and target expressions, suggesting the contribution of the loss in smooth translation of the pixels. Removing L_{qual} and L_{id} deteriorates the overall naturalness, with artifacts manifesting in the eye and mouth regions. As expected, the overall best metrics are obtained when the full LEGAN model with all the loss components is utilized.

7. LEGAN: Optimal Upsampling

To check the effect of the different upsampling approaches on hallucination quality, we separately apply bilinear interpolation, transposed convolution [22] and pixel shuffling [17] on the decoder module of the three hourglass networks in LEGAN's generator G. While the upsampled pixels are interpolated based on the original pixel in the first approach, the other two approaches explicitly learn the possible intensity during upsampling. More specifically, pixel shuffling blocks learn the intensity for the pixels in the fractional indices of the original image (i.e. the upsampled indices) by using a set convolution channels and have been shown to generate sharper results than transposed convolutions. Unsurprisingly, it generates the best quantitative results by outperforming the other two upsampling approaches on 3 out of the 5 objective metrics, as shown in Table 7. Hence we use pixel shuffling blocks in our final implementation of LEGAN.

However, as can be seen in Figure 6, the expression and lighting transformation masks M_e and M_l are more meaningful when interpolated rather than explicitly learned. This interpolation leads to a smoother flow of upsampled pixels with facial features and their transformations visibly more noticeable compared to transposed convolutions and pixel shuffling.

8. LEGAN: Optimal Value of q

As discussed in the main text, we set the value of the hyper-parameter q = 8 for computing the quality loss L_{qual}. We arrive at this specific value after experimenting with different possible values. Since q acts as a target for perceptual quality while estimating L_{qual} during the forward pass, it can typically range from 5 (realistic) to 10 (hyper-realistic). We set q to all possible integral values between 5 and 10 for evaluating the synthesis results both qualitatively (Figure 7) and quantitatively (Table 8).

As can be seen, when q is set to 8, LEGAN generates more stable images with much less artifacts compared to other values of q. Also, the synthesized expressions are visibly more noticeable for this value of q (Figure 7, top row). When evaluated quantitatively, images generated by LEGAN with q = 8 garner the best score for 4 out of 5 objective metrics. This is interesting as setting q = 10 (and not 8) should ideally generate hyper-realistic images and consequently produce the best quantitative scores. We attribute this behavior of LEGAN to the naturalness distribution of the images used to train our quality model Q. Since majority of these images fell in the (7-8) and (8-9) bins, and very few in (9-10) (as shown in Figure 1), Q's representations are aligned to this target. As a result, Q tends to rate hyper-realistic face images (i.e. images with mean naturalness rating between 8 - 10) with a score around 8. Such an example can be seen in the rightmost column of the first row in Figure 4, where Q rates a hyper-realistic StyleGAN generated image [11] as 8.3. Thus, setting q = 8 for L_{qual} computation (using trained Q's weights) during LEGAN training produces the optimal results.

9. LEGAN: Perceptual Study Details

In this section, we share more details about the interface used for our perceptual study. As shown in Figure 8, we ask the raters to pick the image that best matches a target expression and lighting condition. To provide a basis for making judgement, we also share a real image of the same subject with neutral expression and bright lighting condition. However, this is not necessarily the input to the synthesis models for the target expression and lighting generation, as we want to estimate how these models do when the input image has more extreme expressions and lighting conditions. The image order is also randomized to eliminate any bias.

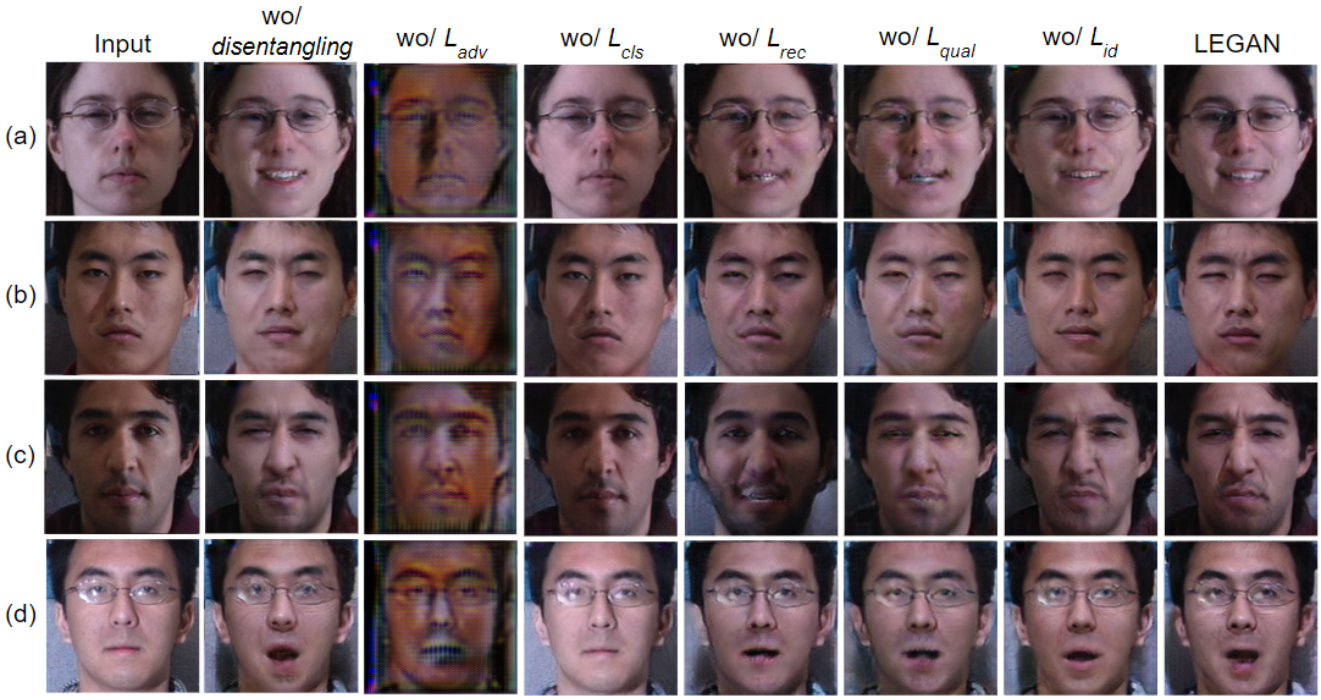

Figure 5: Sample qualitative results from LEGAN and its ablated variants on randomly sampled input images from MultiPIE [7] test set. The target expression and lighting conditions for each row are - (a) (Smile, Left Shadow), (b) (Squint, Ambient), (c) (Disgust, Left Shadow), and (d) (Surprise, Ambient).

Table 6: Ablation studies - quantitative results on held out CMU-MultiPIE [7] test set.

Models	FID [9] ↓	LPIPS [23] ↓	SSIM [19] ↑	Match Score [8, 6] ↑	Quality Score ↑
wo/ disentangling	40.244	0.148	0.557	0.601	5.348
wo/ L_{adv}	351.511	0.460	0.352	0.476	1.74
wo/ L_{cls}	30.236	0.139	0.425	**0.717**	**5.873**
wo/ L_{rec}	40.479	0.135	0.550	0.676	5.475
wo/ L_{qual}	46.420	0.168	0.544	0.621	5.190
wo/ L_{id}	35.429	0.140	0.566	0.587	5.861
LEGAN	**29.964**	**0.120**	**0.649**	0.649	5.853

10. LEGAN: Model Limitations

Although LEGAN is trained on just frontal face images acquired in a controlled setting, it can still generate realistic new views even for non-frontal images with a variety of expressions, as shown in Figures 10 and 11. However, as with any synthesis model, LEGAN also has its limitations. In majority of the cases where LEGAN fails to synthesize a realistic image, the input expression is irregular with non-frontal head pose or occlusion, as can be seen in Figure 9. As a result, LEGAN fails to generalize and synthesizes images with incomplete translations or very little pixel manipulations. One way to mitigate this problem is to extend both our quality model and LEGAN to non-frontal facial poses and occlusions by introducing randomly posed face images during training.

11. LEGAN: More Qualitative Results

In this section, we share more qualitative results generated by LEGAN on unconstrained data from the AFLW [12] and CelebA [13] datasets in Figures 10 and 11 respectively. The randomly selected input images vary in ethnicity, gender, color composition, resolution, lighting, expression and facial pose. In order to judge LEGAN's generalizability, we only train the model on 33k frontal face images from MultiPIE [7] and do not fine tune it on any other dataset.

12. Recolorization Network: Architecture Details

For the colorization augmentation network, we use a generator architecture similar to the one used in [4] for the 128×128×3 resolution. The generator is an encoder-decoder with skip connections connecting the encoder

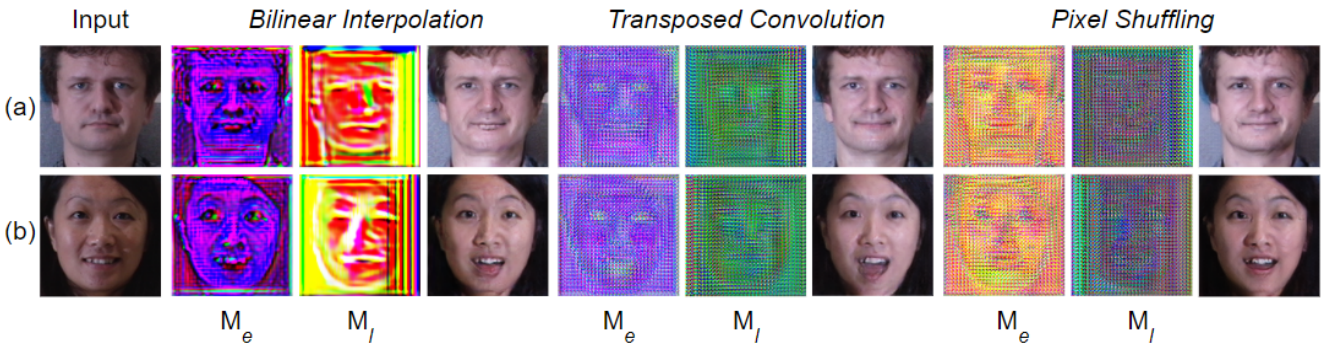

Figure 6: Adding different upsampling techniques in our decoder modules generates hallucinations with slightly different perceptual scores for the same input. Here the target expression and lighting conditions are set as - (a) (Smile, Bright), and (b) (Surprise, Left Shadow). However, the transformation masks M_e and M_l are smoother and more meaningful when bilinear interpolation is used for upsampling. Since both transposed convolution [22] and pixel shuffling [17] learn the intensity of the upsampled pixels instead of simple interpolation, the masks they generate are more fragmented and discrete. We use pixel shuffling in our final LEGAN model.

Table 7: Effects of different upsampling - quantitative results on held out CMU-MultiPIE [7] test set.

Models	FID [9] ↓	LPIPS [23] ↓	SSIM [19] ↑	Match Score [8, 6] ↑	Quality Score ↑
Bilinear Interpolation	29.933	0.128	0.630	**0.653**	5.823
Transposed Convolution [22]	**28.585**	0.125	0.635	0.644	5.835
Pixel Shuffling [17]	29.964	**0.120**	**0.649**	0.649	**5.853**

and decoder layers, and the discriminator is the popular CASIANet [21] architecture. Details about the generator layers can be found in Table 9.

We train two separate versions of the colorization network with randomly selected 10,000 face images from the UMDFaces [5] and FFHQ [10] datasets. These two trained generators can then be used to augment LEGAN's training set by randomly recoloring the MultiPIE [7] images from the training split. Such an example has been shared in Figure 12.

References

[1] DeepFake FaceSwap:. https://faceswap.dev/.

[2] Pre-filtered StyleGAN Images:. https://generated.photos/?ref=producthunt.

[3] S. Banerjee, W. Scheirer, K. Bowyer, and P. Flynn. Fast face image synthesis with minimal training. In *WACV*, 2019. Dataset available here: https://cvrl.nd.edu/projects/data/.

[4] S. Banerjee, W. Scheirer, K. Bowyer, and P. Flynn. On hallucinating context and background pixels from a face mask using multi-scale gans. In *WACV*, 2020.

[5] A. Bansal, A. Nanduri, C. D. Castillo, R. Ranjan, and R. Chellappa. Umdfaces: An annotated face dataset for training deep networks. *IJCB*, 2017.

[6] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman. Vggface2: A dataset for recognizing faces across pose and age. In *arXiv:1710.08092*.

[7] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker. Multi-pie. *Image and Vision Computing.*, 28(5):807–813, 2010.

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

[9] 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.

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

[11] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. In *arXiv:1812.04948*, 2018.

[12] M. Koestinger, P. Wohlhart, P.M. Roth, and H. Bischof. Annotated Facial Landmarks in the Wild: A Large-scale, Real-world Database for Facial Landmark Localization. In *First IEEE International Workshop on Benchmarking Facial Image Analysis Technologies*, 2011.

[13] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *ICCV*, 2015.

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

[15] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner. FaceForensics++: Learning to detect manipulated facial images. In *ICCV*, 2019. Available here: https://github.com/ondyari/FaceForensics.

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

[17] W. Shi, J. Caballero, F. Huszar, J. Totz, A.P. Aitken, R. Bishop, D. Rueckert, and Z. Wang. Real-time single im-

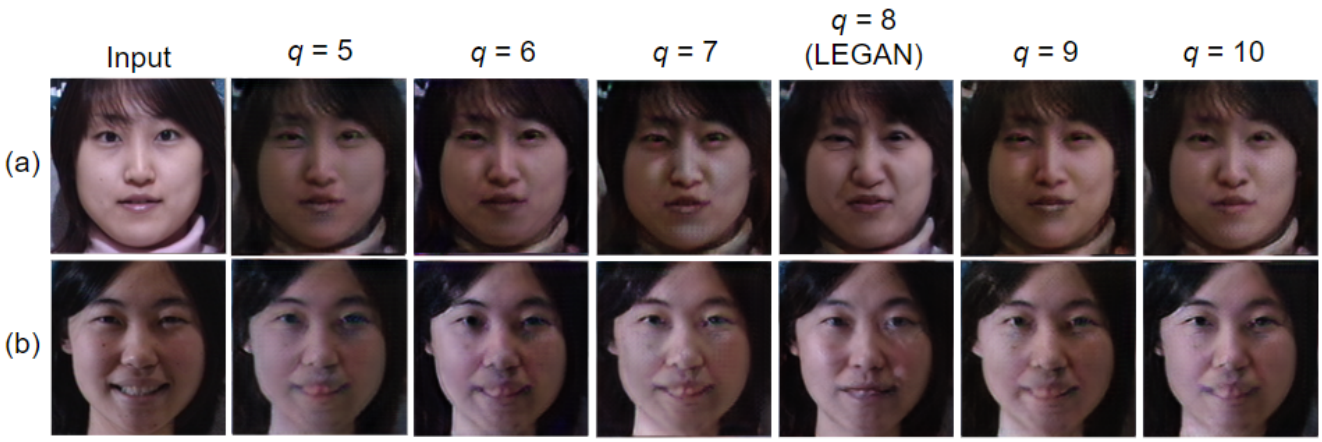

Figure 7: Sample results illustrating the effect of the hyper-parameter q on synthesis quality. The input images are randomly sampled from the MultiPIE [7] test set with target expression and lighting conditions set as - (a) (Disgust, Left Shadow), and (b) (Neutral, Right Shadow). Since it generates more stable and noticeable expressions with fewer artifacts, we set $q = 8$ for the final LEGAN model.

Table 8: Quantitative results on the held out CMU-MultiPIE [7] test set by varying the value of the hyper-parameter q.

Models	FID [9] ↓	LPIPS [23] ↓	SSIM [19] ↑	Match Score [8, 6] ↑	Quality Score ↑
q = **5**	41.275	0.143	0.550	0.642	5.337
q = **6**	44.566	0.139	0.542	0.651	5.338
q = **7**	38.684	0.137	0.631	**0.663**	5.585
q = **8 (LEGAN)**	**29.964**	**0.120**	**0.649**	0.649	**5.853**
q = **9**	42.772	0.137	0.637	0.659	5.686
q = **10**	46.467	0.132	0.586	0.583	5.711

Table 9: Colorization Generator 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

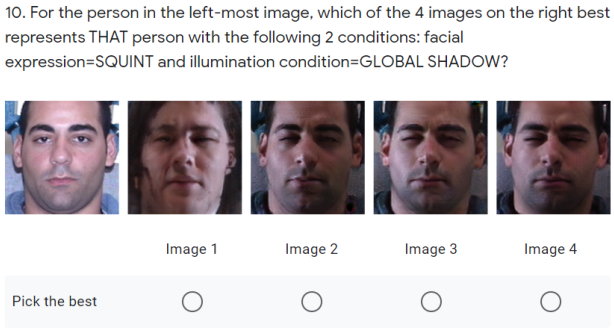

Figure 8: Our perceptual study interface: given a base face image with neutral expression and bright lighting (leftmost image), a rater is asked to select the image that best matches the target expression ('Squint') and lighting ('Right Shadow') for the same subject.

age and video super-resolution using an efficient sub-pixel convolutional neural network. In *CVPR*, 2016.

[18] D. Ulyanov, A. Vedaldi, and V. Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv:1607.08022*, 2016.

[19] 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.

[20] B. Xu, N. Wang, T. Chen, and M. Li. Empirical eval-

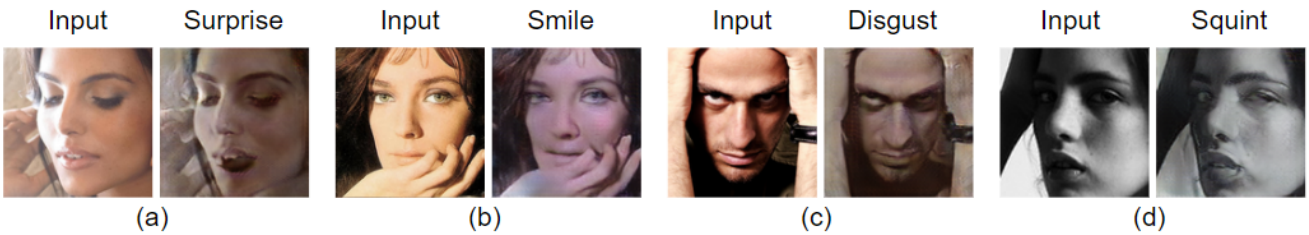

Figure 9: Failure cases: for each input image LEGAN fails to correctly generate the target facial expression. In (a) LEGAN manages to generate a surprised mouth but fails to open the subject's eyes, (b) the smile is half generated due to occlusion by the subject's fingers, (c) the target disgusted expression is missing, and (d) the subject's eyes are not squinted. Most of these failure cases are either due to non-frontal facial pose or occlusion.

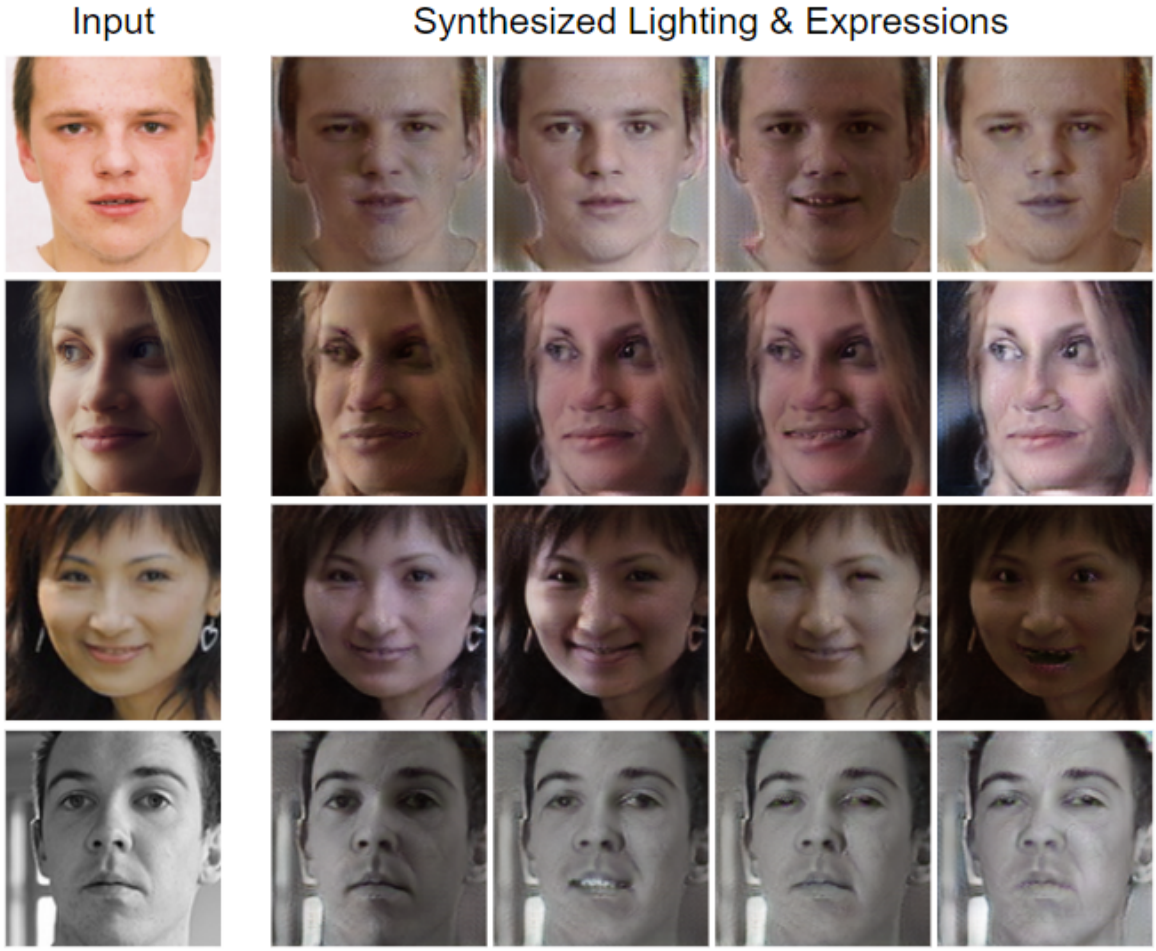

Figure 10: Synthesized expressions and lighting conditions for the same input image, as generated by LEGAN. These input images are randomly selected from the AFLW [12] dataset and the results are generated by randomly setting different expression and lighting targets. LEGAN is trained on 33k frontal face images from MultiPIE [7] and we do not fine-tune the model on any other dataset. All images are $128 \times 128 \times 3$.

uation of rectified activations in convolutional network. *arXiv:1505.00853*, 2015.

[21] D. Yi, Z. Lei, S. Liao, and S. Z. Li. Learning face representation from scratch. In *arXiv:1411.7923*.

[22] M.D. Zeiler, D. Krishnan, G.W. Taylor, and R. Fergus. Deconvolutional networks. In *CVPR*, 2010.

[23] R. Zhang, P. Isola, A.A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

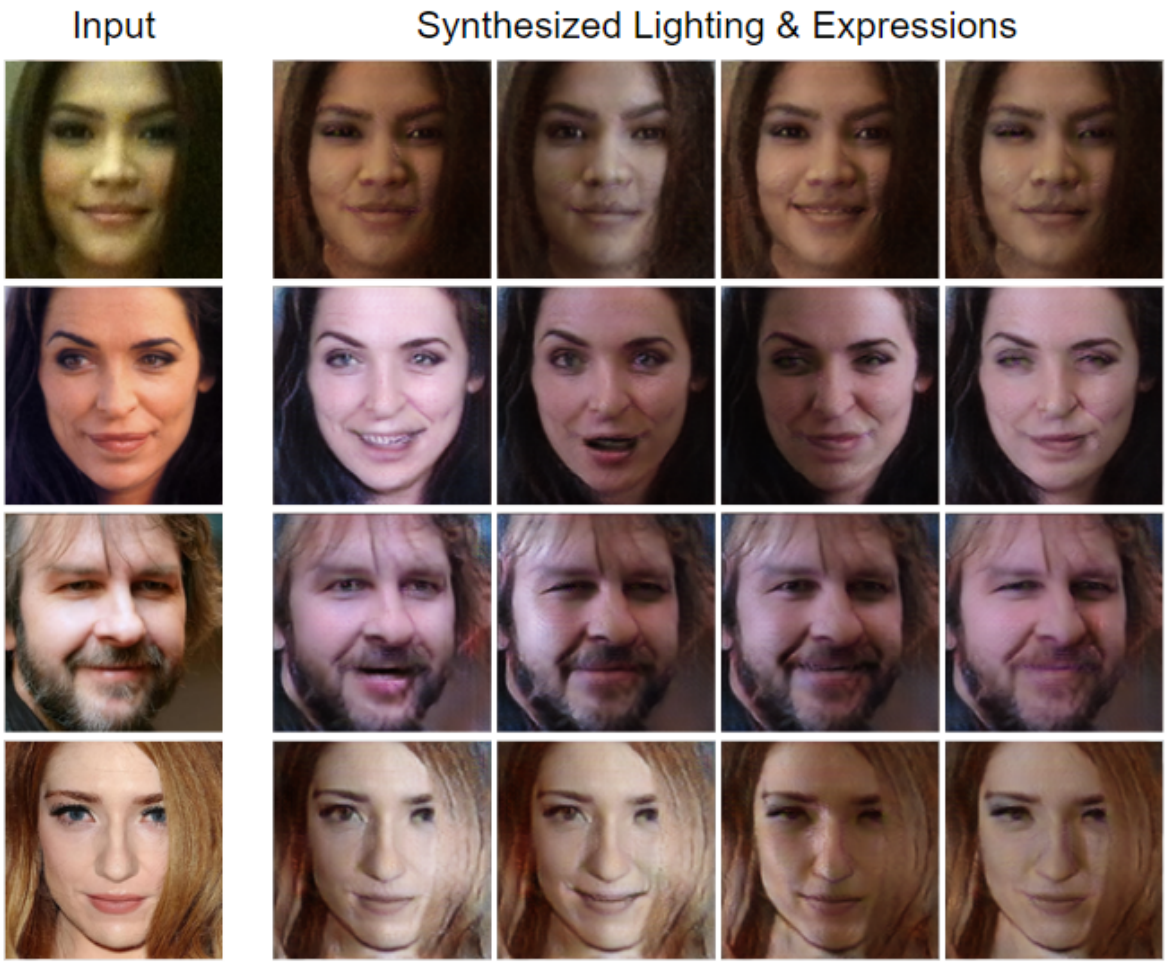

Figure 11: Synthesized expressions and lighting conditions for the same input image, as generated by LEGAN. These input images are randomly selected from the CelebA [13] dataset and the results are generated by randomly setting different expression and lighting targets. LEGAN is trained on 33k frontal face images from MultiPIE [7] and we do not fine-tune the model on any other dataset. All images are $128 \times 128 \times 3$.

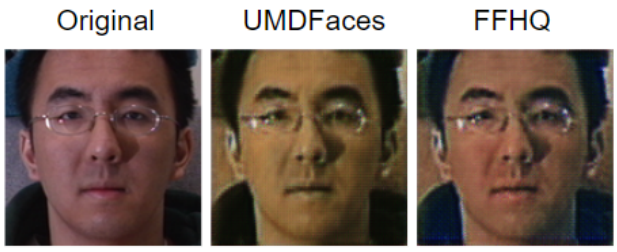

Figure 12: Recolorization Example: We randomly select a test image from the MultiPIE [7] dataset and recolor it using the colorization generator snapshots, trained using UMDFaces [5] and FFHQ [10] datasets respectively. Although the image is recolored, its lighting is preserved by the colorization generator.