Supplementary for "Property Tunable Image Manipulation with Self-Generated Supervision"

Anonymous WACV submission

Paper ID 697

1. Implementation Details

For the invariant encoder E, we adopt three down-sample blocks. Each down-sample block contains a Conv-InstanceNorm-ReLU module, and the kernel size and stride of the convolution operation are 3 and 2, separately. The shape of the invariant feature representation of E is $B \times 32 \times 32 \times 256$, where B is the batch size and we set $B = 1$ in our experiments. For the property decoder P, we adopt 5 up-sample blocks to obtain the tunable property feature representation, which has the same shape as the invariant feature representation. The up-sample block contains the Deconv-InstanceNorm-ReLU module (the kernel size and stride are also 3 and 2, respectively). For the invariant capsule C, we refer to the architecture design of [8], which adopts the dynamic routing to improve the ability to obtain disentanglement. For the image generator G, we first use 9 residual blocks to fuse the tunable and invariant property information, for the residual blocks, we adopt the same architecture as CycleGAN [11] and the stride is 1. Then we use three Deconv-Instance-ReLU modules (the kernel size and stride are also 3 and 2, respectively) to get the same size image output, followed with one Tanh activity function. For the multiple-branch estimator and discriminator, the shared backbone contains 3 Conv-LeakyReLU modules, the kernel size and stride are 4 and 2, separately. For the LeakyReLU operation, the threshold α is set as 0.01. The estimator branch has the same architectural design as the invariant capsule C. The discriminator branch adopts the Patch-GAN architectural design of Pix2pix [3], and we design one 4×4 patch output for every branch. Please note, we adopt the least square loss [7] to compute the adversarial loss. For the optimization, we adopt Adam optimizer [4] to optimize the final objective function with the learning rate 0.0002.

2. Recognition effectiveness of the invariant feature

We also explored the effectiveness of the invariant feature of the invariant capsule branch, we adopt the invariant feature to boost the face recognition performance on FEI [9] dataset. First, we perform a Siamese network [10] to achieve face recognition. In order to verify that invariant feature representation from TunerGAN can help boost the performance of facial recognition, we design two experiments where we utilize the Siamese network for binary classification to decide if the two images came from the same person. To ensure the fairness of the experiment, we use the invariant encoder of TunerGAN as the backbone of the Siamese network to acquire a feature map that has the same dimension as the invariant feature representation ($1 \times 32 \times 32 \times 256$). Then we apply several Conv-ReLU layers feature map to obtain the Siamese code from the image. In this setting, the Siamese codes from two different images are used to determine whether the two original images are from the same identity. Using the baseline model described above, we trained using the images from 160 different people, each identity with 10 different poses. Positive and negative training examples are sampled randomly during the stage. At the evaluation stage, we generate 200 positive and 200 negative samples from 40 people that are unseen during the training time. (We generate 5 positive pairs from each of the 40 people.) The baseline model classified 372 out of the 400 test examples correctly, giving an accuracy of 93%. Then for comparison, we run the experiments with TunerGAN network. We concatenate the trained invariant feature from TunerGAN with the feature representation from the Siamese backbone to generate the Siamese code. Different from the original baseline, the invariant feature representation also has 256 channels, so the current design has 512 channels with more network parameters. In this setup, we train our TunerGAN and Siamese Network simultaneously. Using the same test examples, the model with our TunerGAN is able to classify 379 of them correctly, giving an accuracy of 94.75% that is almost 2% higher than the baseline model, which indicates that our invariant feature can boost the recognition performance.

Table 1. Quantitative comparison of different settings on FEI dataset.

Method	Without Invariant feature	With Invariant feature
Accuracy	93.00%	94.75%

3. Comparison between TunerGAN and StarGAN+MUNIT

To better illustrate the difference between our TunerGAN and the combination of StarGAN and MUNIT, we perform the visual comparison on FEI dataset, and the visual and quantitative comparison is shown in Figure 1 and Table 2. We also report the result of the raw StarGAN method. With self-guided supervision, our TunerGAN can generate more precise images for the pose translation task. Due to the infinite label sampling, our method can go beyond the existing sample distribution (generating data for nonexistent labels). With these generated samples, our TunerGAN model can be further trained with such extra data-label pairs and obtain performance improvement.

Figure 1. The visual generation comparison between StarGAN, StarGAN+MUNIT and our TunerGAN. Our TunerGAN can generate more realistic images with more detailed information.

4. Comparison on emotional expression

In this section, we The comparison between other methods use the data and training setting is the same as the head pose rotation. The visual comparison and quantitative comparison are shown in Figure 2 and Table 3. As the results in head pose rotation, we outperform the StarGAN and STGAN trained on un-paired data and get a comparable result on with Pix2pix trained on paired data.

5. Comparison on foggy scene dataset

We have tested on other tasks on scene datasets. Here we show an example of applying it to fog thickness tuning on the Foggy Kitti dataset, in comparison with StarGAN and STGAN. Foggy Kitti [2] provides 7 categories fog images with different visibility: 30m, 40m, 50m, 75m, 150m, 375m, 750m. We follow the train/test split. Table 4 clearly shows TunerGAN significantly outperforms other methods in all metrics. And the visual comparison of this dataset is shown in Figure 3, we omit the input image with 375m visibility. As shown, our TunerGAN can generate more plausible images with detailed information.

References

[1] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation. In *CVPR*, pages 8789–8797, 2018. 3

[2] Shirsendu Sukanta Halder, Jean-François Lalonde, and Raoul de Charette. Physics-based rendering for improving robustness to rain. In *IEEE/CVF International Conference on Computer Vision*, 2019. 2

[3] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, pages 5967–5976, 2017. 1, 3

[4] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 1

[5] Ming Liu, Yukang Ding, Min Xia, Xiao Liu, Errui Ding, Wangmeng Zuo, and Shilei Wen. Stgan: A unified selective transfer network for arbitrary image attribute editing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3673–3682, 2019. 3

[6] Liqian Ma, Xu Jia, Qianru Sun, Bernt Schiele, Tinne Tuytelaars, and Luc Van Gool. Pose guided person image generation. In *Advances in Neural Information Processing Systems*, pages 406–416, 2017. 3

[7] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *ICCV*, pages 2794–2802, 2017. 1

[8] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In *Advances in Neural Information Processing Systems*, pages 3859–3869, 2017. 1

[9] Carlos Eduardo Thomaz and Gilson Antonio Giraldi. A new ranking method for principal components analysis and its application to face image analysis. *Image and Vision Computing*, 28(6):902–913, 2010. 1

[10] Weihong Wang, Jie Yang, Jianwei Xiao, Sheng Li, and Dixin Zhou. Face recognition based on deep learning. In *International Conference on Human Centered Computing*, pages 812–820. Springer, 2014. 1

[11] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *International Conference on Computer Vision*, pages 2223–2232, 2017. 1

Table 2. Quantitative comparison of emotional translation on RaFD dataset. The symbol ↑ (↓) indicates that the larger (smaller) the value, the better the performance. † represents the supervised methods.

Method	SSIM↑	PSNR↑	MSE↓	RMSE↓	LPIPS↓	FID↓	Classification Acc↑
StarGAN	0.8812	15.17	0.0247	0.1512	0.2567	93.26	0.9325
StarGAN+MUNIT	0.9015	15.27	0.0236	0.1501	0.2418	78.22	0.9375
TunerGAN	**0.9219**	**16.68**	**0.0200**	**0.1288**	**0.2123**	**66.56**	**0.9650**

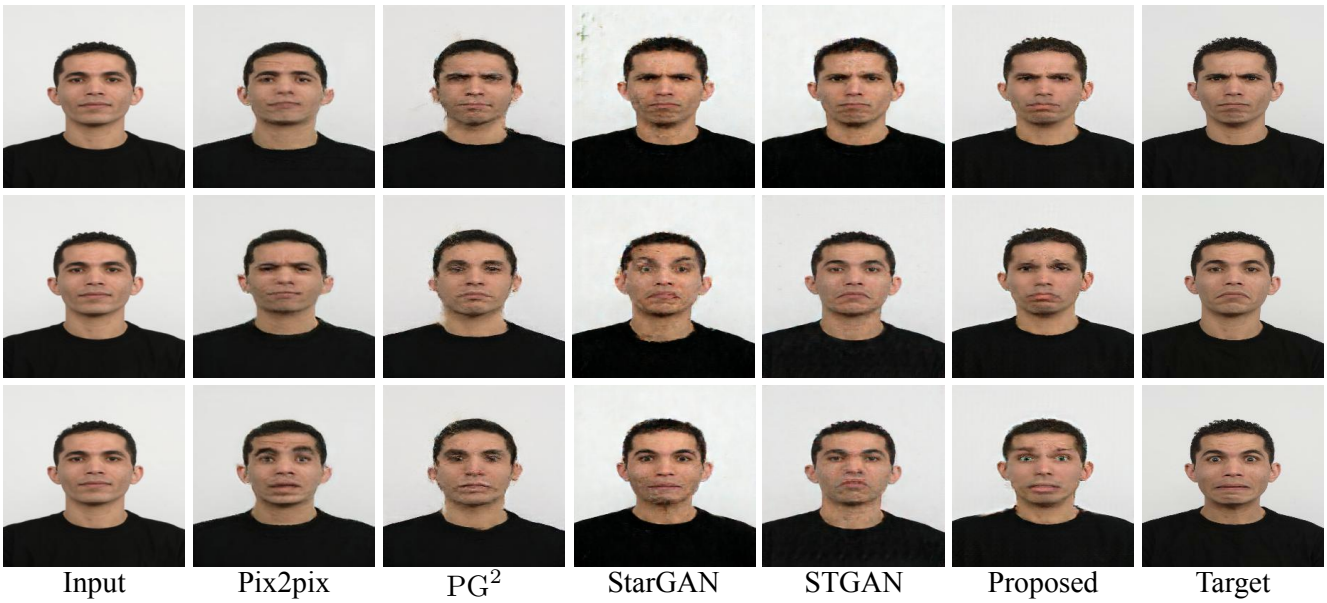

Figure 2. The visual comparison of using both supervised and unsupervised methods on RaFD dataset.

Table 3. Quantitative comparison of emotional translation on RaFD dataset. The symbol ↑ (↓) indicates that the larger (smaller) the value, the better the performance. † represents the supervised methods.

Method	SSIM↑	PSNR↑	MSE↓	RMSE↓	LPIPS↓	FID↓	Classification Acc↑
Pix2pix† [3]	0.9697	20.6510	0.01047	**0.0978**	0.1135	54.9618	0.9583
PG2† [6]	0.9671	20.2226	**0.01026**	0.0990	**0.1101**	64.4190	**0.9895**
StarGAN [1]	0.8430	19.0748	0.01434	0.1155	0.1401	128.3860	0.9583
STGAN [5]	0.9448	20.5620	0.01197	0.1030	0.1341	90.0433	0.9687
Proposed	**0.9704**	**21.0534**	0.01036	0.0993	0.1142	**52.6346**	0.9791

Table 4. Quantitative comparison of fog thickness on Foggy Kitti dataset. The symbol ↑ (↓) indicates that the larger (smaller) the value, the better the performance.

Method	SSIM↑	PSNR↑	MSE↓	RMSE↓	LPIPS↓	FID↓	Classification Acc↑
StarGAN	0.9713	33.74	0.00041	0.0171	0.0641	36.19	0.9736
STGAN	0.9746	34.17	0.00033	0.0162	0.0612	23.87	0.9824
TunerGAN	**0.9811**	**35.89**	**0.00025**	**0.0154**	**0.0587**	**18.76**	**0.9876**

Figure 3. The visual comparison of tuning fog thickness on Foggy Kitti dataset. We omit the input image with 375m visibility.