VecGAN: Image-to-Image Translation with Interpretable Latent Directions

Yusuf Dalva, Said Fahri Altındiş, and Aysegul Dundar

Bilkent University
{yusuf.dalva, fahri.altindis}@bilkent.edu.tr
adundar@cs.bilkent.edu.tr

A More comparisons

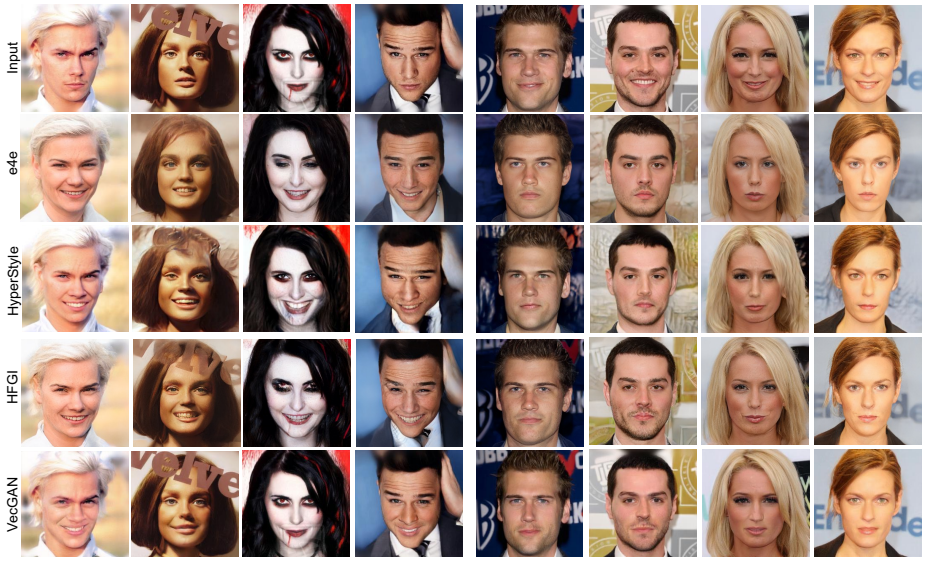

Fig. 1: Qualitative results of smile attribute of our model (VecGAN) and other StyleGAN based models.

In Fig. 1, we compare our method with other methods that are proposed to invert images to StyleGANv2 space and perform edits via the pretrained StyleGANv2. We compare with e4e [4], HyperStyle [2], and HFGI [5]. Same input examples are used from Fig. 5 main paper. e4e as also stated in their paper outputs results with worse distortion (input-output similarity) but better edits. HyperStyle and HFGI are concurrent works with improved fidelity to the input image but still significantly worse than our method both in edit quality and reconstruction quality of the input details.

Fig. 2: Qualitative results of smile attribute of our model (VecGAN) and other StyleGAN based editing models.

We additionally compare with StyleFlow [1] and StyleSpace [6] in Fig. 2. For both examples, we take their real image editing example from their papers and feed the input crops to VecGAN for comparison. As can be seen from Fig. 2, both methods suffer from the limitations of the projection method as inputs are not faithfully reconstructed. Additionally, the edit is not perfectly disentangled in StyleFlow example as the strap of the top changes when smile is modified. VecGAN achieves significantly better results in these examples.

B Additional Quantitative Results

Method	KID(+)	KID(-)	KID (Avg)
L2M-GAN	0.01010	0.00942	0.00976
InterfaceGAN	0.00603	0.00671	0.00637
VecGAN	**0.00188**	**0.00328**	**0.00258**

Table 1: Quantitative results for Setting B - Smile attribute.

In Table 1, we compare VecGAN and other competing methods with KID metric [3]. Same as in FID evaluation, VecGAN achieves significantly better results.

C Model Architecture

In this section, we provide architectural details of VecGAN.

Generator. Our generator is composed of an encoder and decoder. For encoder, we use 8 successive blocks that perform downsampling which reduce feature map dimensions to 1x1. In our decoder, we have an architecture symmetric to encoder, which is composed of 8 successive upsampling blocks. Except the last downsampling block and the first upsampling block, we use instance normalization denoted as (+IN). The channels increase as {64, 64, 128, 256, 512, 512, 512, 1024, 2048} (for output resolution 256x256) in the encoder and decrease in a symmetric way in the decoder. In addition to these building blocks, we use a skip connection between the encoder and decoder as shown in Fig. 3.

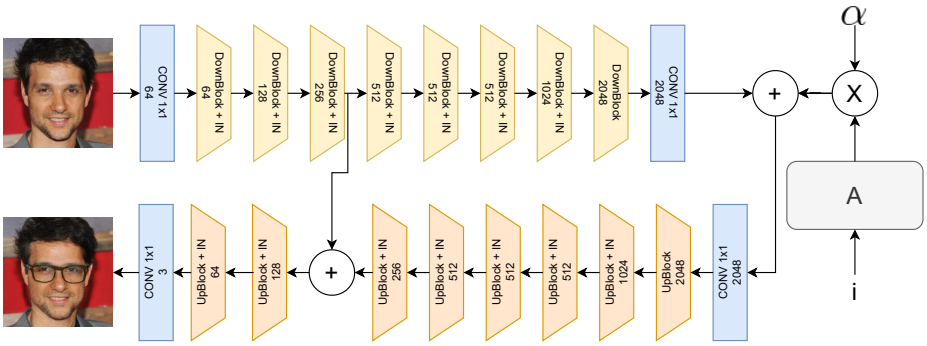

Fig. 3: Generator architecture. Numbers correspond to the output channels of each block.

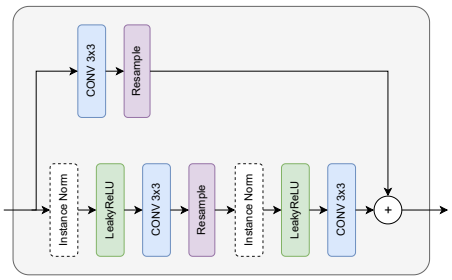

Fig. 4: Architecture for the residual blocks.

Residual Blocks. Each DownBlock and UpBlock has a residual block with $3x3$ convolutional filters followed by a downsampling and upsampling layer, respectively. For downsampling, we use average pooling and for upsampling, we use nearest-neighbor. We use LeakyReLU activation layer and instance normalization layer in each convolutional module.

Discriminator. Discriminator also employs an architecture with decreasing resolution and increasing channel size as given in Fig. 5. Just like the generator, we build our discriminator with channel sizes of {64, 64, 128, 256, 512, 512, 512, 1024, 2048}, that reduces the feature map dimensions to 1x1. At the end, we concatenate the extracted style α_t from the input image to this latent code and apply a 1x1 convolution. This final convolution is specific to each tag-attribute pair so that the model can use this information.

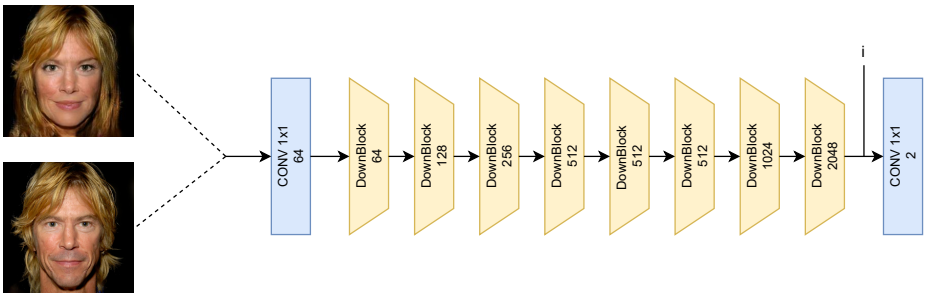

Fig. 5: Architecture of the discriminator. Discriminator takes an input image and processes it with downsampling blocks with increased number of channels. Towards the end, the extracted feature map with 1x1 feature dimensions is concatenated with the scale of the input image. As we perform scale extraction for the image in the cycle-translation path, no additional scale extraction is needed.

Hyperparameters. For training our framework, we set the following parameters; $\lambda_a = 1$, $\lambda_{rec} = 1.5$, $\lambda_s = 1$, $\lambda_o = 1$ and $\lambda_{sp} = 0.05$. We use a learning rate of 10^{-4} and train our model for 500K iterations with a batch size of 8 on a single GPU. For the feature encoding and feature directions in matrix A, we use a 2048 dimensional vector representation same as the channel size of the last convolutional layer from the encoder.

D Additional Results

We provide additional qualitative results of our method in Fig. 6, 7, 8, 9, 10, and 11.

Fig. 6: Smile tag manipulation results. First and third rows show input images. Second and forth rows show image translation results.

Fig. 7: Glasses tag manipulation results. First and third rows show input images. Second and forth rows show image translation results.

Fig. 8: Gender tag manipulation results. First and third rows show input images. Second and forth rows show image translation results.

Fig. 9: Bangs tag manipulation results. First and third rows show input images. Second and forth rows show image translation results.

Fig. 10: Age tag manipulation results. First and third rows show input images. Second and forth rows show image translation results.

Fig. 11: Hair tag manipulation results. First and third rows show input images. Second and forth rows shows image translation results.

References

1. Abdal, R., Zhu, P., Mitra, N.J., Wonka, P.: Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows. ACM Transactions on Graphics (ToG) **40**(3), 1–21 (2021)
2. Alaluf, Y., Tov, O., Mokady, R., Gal, R., Bermano, A.: Hyperstyle: Stylegan inversion with hypernetworks for real image editing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18511–18521 (2022)
3. Bińkowski, M., Sutherland, D.J., Arbel, M., Gretton, A.: Demystifying mmd gans. arXiv preprint arXiv:1801.01401 (2018)
4. Tov, O., Alaluf, Y., Nitzan, Y., Patashnik, O., Cohen-Or, D.: Designing an encoder for stylegan image manipulation. ACM Transactions on Graphics (TOG) **40**(4), 1–14 (2021)
5. Wang, T., Zhang, Y., Fan, Y., Wang, J., Chen, Q.: High-fidelity gan inversion for image attribute editing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11379–11388 (2022)
6. Wu, Z., Lischinski, D., Shechtman, E.: Stylespace analysis: Disentangled controls for stylegan image generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12863–12872 (2021)