Supplementary Materials of Fine-grained Image-to-Image Transformation towards Visual Recognition

Wei Xiong[1] Yutong He[1] Yixuan Zhang[1] Wenhan Luo[2] Lin Ma[2] Jiebo Luo[1]
[1]University of Rochester [2]Tencent AI Lab
[1]{wxiong5,jluo}@cs.rochester.edu, yhe29@u.rochester.edu, yzh215@ur.rochester.edu
[2]{whluo.china, forest.linma}@gmail.com

1. Structure of Our Model

In this section, we provide the detailed structure of our model, including the structure of our generators, our discriminators and adaptive identity modulation module in our generator.

Table 1. The architecture of the generator for CompCars dataset. The size of the input image is $3 \times 224 \times 224$.

Layer	#Channels	Kernel size	Stride	Padding
Conv1	32	3	1	1
Conv2	64	4	2	1
Conv3	128	4	2	1
Conv4	128	4	2	1
Conv5	256	4	2	1
Conv6	256	4	2	1
Conv7	512	3	2	1
Conv8	512	4	1	0
DConv1	512	4	1	0
DConv2	256	3	2	1
DConv3	256	4	2	1
DConv4	128	4	2	1
DConv5	128	4	2	1
DConv6	64	4	2	1
DConv7	32	4	2	1
DConv8	3	1	1	0

1.1. Generator on CompCars Dataset

The generator on CompCars dataset is composed of an encoder and a decoder. Given an image of size 224×224, the encoder maps the image to a vector of identity with size 512×1. Then the identity feature vector is concatenated with an attribute condition code C with size 5×1 and a random noise vector z with size 128×1, to form a latent vector. The latent vector is then decoded by the decoder with several deconvolution layers. Each convolution layer is followed by a batch normalization layer and a *Leaky ReLU* layer [1], except *Conv1*, *Conv8*, *DConv1* and *DConv8* layers. We use *Tanh* function in the final layer. We use constrained nonalignment connection to link feature maps of *Conv4* and *DConv4*, which have a spatial size of 28×28 or feature maps of *Conv3* and *DConv5*, which have a spatial size of 56×56. The detailed structure of the generator for CompCars dataset is present in Table 1.

1.2. Discriminator on CompCars Dataset

The discriminator used on CompCars dataset has a similar architecture as the encoder of our generator. It is composed of several convolution layers, followed by a global average pooling layer and two classification layers. Each convolution layer is followed by a batch normalization layer and a *Leaky ReLU* layer, except *Conv1* layer. The detailed structure is shown in Table 3.

1.3. Generator on Multi-PIE Dataset

Regarding the generator on Multi-PIE dataset, for a fair comparison, we take the architecture of DR-GAN [3] as our basic architecture. The generator takes a 96×96 image, a random noise with size 128×1, and a viewpoint condition code with size 9×1 as input, and outputs a 96×96 image. The only difference from the original DR-GAN is that we do not take illumination as a condition. We use the same setting for DR-GAN in our experiments. We apply our proposed constrained nonalignment connection method to link feature maps with a spatial size of 24×24. We also modulate the feature maps in the decoder with the identity feature.

1.4. Discriminator on Multi-PIE Dataset

For a fair comparison, we use a similar discriminator architecture for Multi-PIE dataset as that used by DR-GAN [3]. The only difference is that we do not classify the illumination of the images. Such a setting is kept the same for DR-GAN when we compare our model with DR-GAN.

1.5. Adaptive Identity Modulation

In this paper, we propose an adaptive identity modulation (AIM) method, to integrate identity information into

Table 2. Identity preservation experiment results with different radius of our model (vanilla+CNC(28)) on CompCars dataset. Experiments are carried out with 20, 50, 80, 120 and 200 categories from the standard set. We report both top-1 and top-5 accuracies. % is omitted for convenience.

model	20c-top1	20c-top5	50c-top1	50c-top5	80c-top1	80c-top5	120c-top1	120c-top5	200c-top1	200c-top5
CNC, r=3	52.34	75.93	38.49	59.34	29.66	50.80	23.09	43.02	18.51	35.81
CNC, r=7	**55.05**	**80.16**	**42.24**	**63.49**	**34.68**	**56.09**	**26.50**	46.44	**22.70**	**40.56**
CNC, r=11	53.70	78.88	39.62	60.68	32.48	54.40	26.11	**46.71**	20.27	38.16
CNC, r=14	53.12	77.08	38.30	59.12	30.40	52.13	23.82	44.01	18.13	34.84

Table 3. The architecture of the discriminator for CompCars dataset. The size of the input image is $3 \times 224 \times 224$.

Layer	#Channels	Kernel size	Stride	Padding
Conv1	32	4	2	1
Conv2	64	4	2	1
Conv3	128	4	2	1
Conv4	256	4	2	1
Conv5	256	4	2	1
GAP	-	-	-	-
Linear	1181	-	-	-
Linear	5	-	-	-

the convolutional feature blocks in a more effective way. Here we provide the detailed structures of the sub-modules in AIM. First, the attention vector att_B is obtained by mapping the average feature B_f with an attention layer. The attention layer is composed of a linear layer with C_{id} output nodes and a Sigmoid function, where C_{id} is the number of feature points in the identity feature f_{id}. Second, after we obtain the attended identity feature $f_{id}^{att} = f_{id} \odot att_B$, we need to map it to γ and β. Specifically, we map f_{id}^{att} to γ by a multi-layer perceptron (MLP) of two layers, where the hidden layer has 256 nodes with ReLU activation function, and the last layer is a linear layer without any activation function. Similarly, we map f_{id}^{att} to β with another two-layer MLP with the same configuration. These two MLPs do not share weights. Third, we use the modulated γ and β to re-scale the normalized feature map $\hat{B}_i$, to obtain the final feature map $\tilde{B}_i = \gamma(f_{id}, B_i)\hat{B}_i + \beta(f_{id}, B_i)$.

2. Ablation Study on Radius r in CNC

In the identity preservation experiment on CompCars with N_c classes, where $N_c = 20, 50, 80, 120, 200$, we set the radius of our proposed constrained nonalignment connection (CNC) r to be 3, 7, 11 and 14 on model "vanilla + CNC(28)". Note that when $r = 14$, our CNC model degrades to the Global-NC version, as when calculating the non-local attention, it searches over all possible locations in the key space. The classification results are shown in Table 2. From the results, we conclude that r plays a key role in our model and has a significant influence on the final performance. When $r = 3$, the structure of our CNC module is close to the traditional skip-connection. In such a situation, the feature in the decoder matches only very few feature points in the encoder feature, ignoring the global dependencies. As such, the model with $r = 3$ fails to capture sufficient contextual details from the encoder feature. On the contrary, our model with $r = 14$ matches the feature point in the decoder to all locations of the encoder feature, which does not achieve the best performance. The reason may be that matching over all locations in the encoder feature can be difficult to optimize, and is vulnerable to the noise in the feature map. Results in Table 2 show that when $r = 7$, our model achieves the best performance.

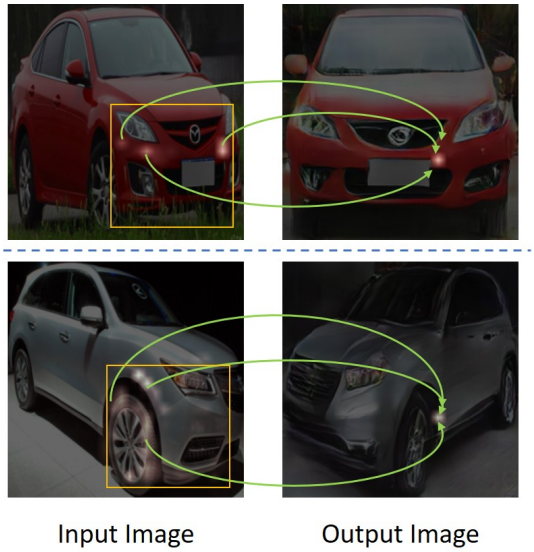


Figure 1. Two examples of learned attention weights on CompCars dataset. In each example, the images in the left and right are input image and output image, respectively. We select a pixel in the output image (indicated by the white point) and visualize the learned constrained attention for this pixel, as shown in the left image. Specifically, the left image shows the salient locations (indicated by the white regions) of the input image attended by the selected pixel in the output image. The yellow rectangle refers to the neighborhood region that the selected pixel needs to attend. The green arrows denote the constrained nonalignment connection that links the attended pixels in the input image with the pixel in the output image.

3. Attention Visualization

In this section, we visualize the attention learned by the generators on CompCars dataset. In Fig. 1, we show the

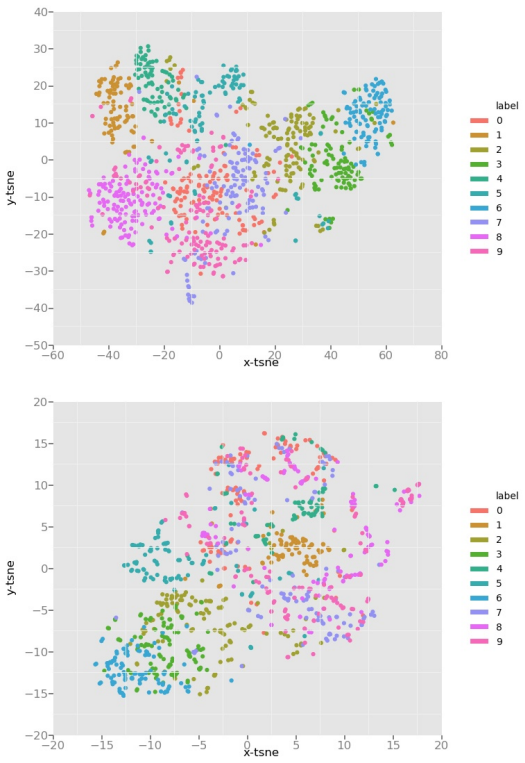
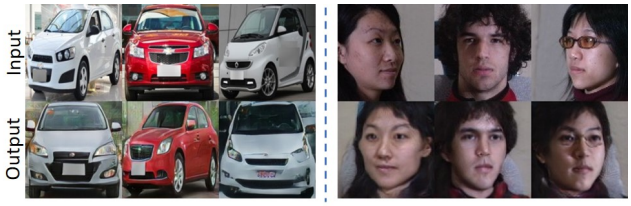


Figure 2. Distribution of images generated by our model (top) and DR-GAN (bottom), visualized with t-SNE. Our model produces samples with more evident cluster property.

attention map learned for a fixed point in the output image. From the figures, pixel of the bumper in the output image can attend to the pixels of the bumper in the input image. Pixel of the frontal tier in the output image can correctly attend to regions around the tier in the input image. Results demonstrate that the attention can be well learned so that the pixel in the output image can correctly attend to regions in the input image that are relevant to the output pixel.

4. Visualization of Data Distribution (t-SNE)

We further compare each model by visualizing the t-Distributed Stochastic Neighbor Embedding (t-SNE) [2] plot of their generated images from CompCars dataset. We only compare our model with DR-GAN, since it has the best identity preservation ability among all the models we have compared. We generate images of 10 classes with our model/DR-GAN and use the resnet18 model to extract features of each image, then plot the images. As can be seen from Fig. 2, our model can generate categorical data with smaller intra-class variance and larger inter-class variance compared to DR-GAN, further demonstrating the superiority of our model on identity preservation.



(a) Failure results on CompCars (b) Failure results on Multi-PIE

Figure 3. Failure cases of our model on CompCars and Multi-PIE datasets. Top: input images, bottom: generated images.

5. More Results on Multi-PIE

In this section, we show more images generated from the test set of Multi-PIE dataset. As shown in Fig. 4, faces generated by our model look more similar to the input face and the ground-truth face in terms of identity, while faces generated by the existing models differ from the input face both in the overall identity and in many specific details, such as the shape of the jaw, hair style, mouth. The results are consistent with the classification performance of each model. Note that in some examples, the image generated with the same viewpoint as the input image may not be exactly the same as the input image. The results of our model is reasonable, as in our task, we do not expect the output image to reconstruct the input image. Our model allows for diversity of images while preserving the identity so that the generated images can better augment the dataset.

6. Failure Cases

In this section, we show several failure cases of our model on CompCars and Multi-PIE datasets. As shown in Fig. 3 (a), on CompCars, our model fails to maintain some important details of the input image, such the frontal lights and the logos. The failure may be due to the difficulty of our task. Cars in our task contain many fine-grained details, making the task very challenging. Moreover, the images in our dataset are not well-aligned. On Multi-PIE, as shown in the first column of Fig. 3 (b), there are slight distortions on the contour of the face. In the second column of Fig. 3 (b), the jaw shape of the generated face does not highly accord with the input face. In the third column of Fig. 3 (b), our model is able to generate a visually-pleasing face with a pair of glasses, but the glasses exhibit a different style from the input image. In these cases, the faces are not generated perfectly. However, our model can still generate visually pleasing faces that look similar to the input face. The results on both datasets indicate that although our model does not succeed in preserving all the essential identity information in challenging situations, it is still able to capture most of the contextual details. In the future, we will continue to improve the model to maintain more identity-related details.

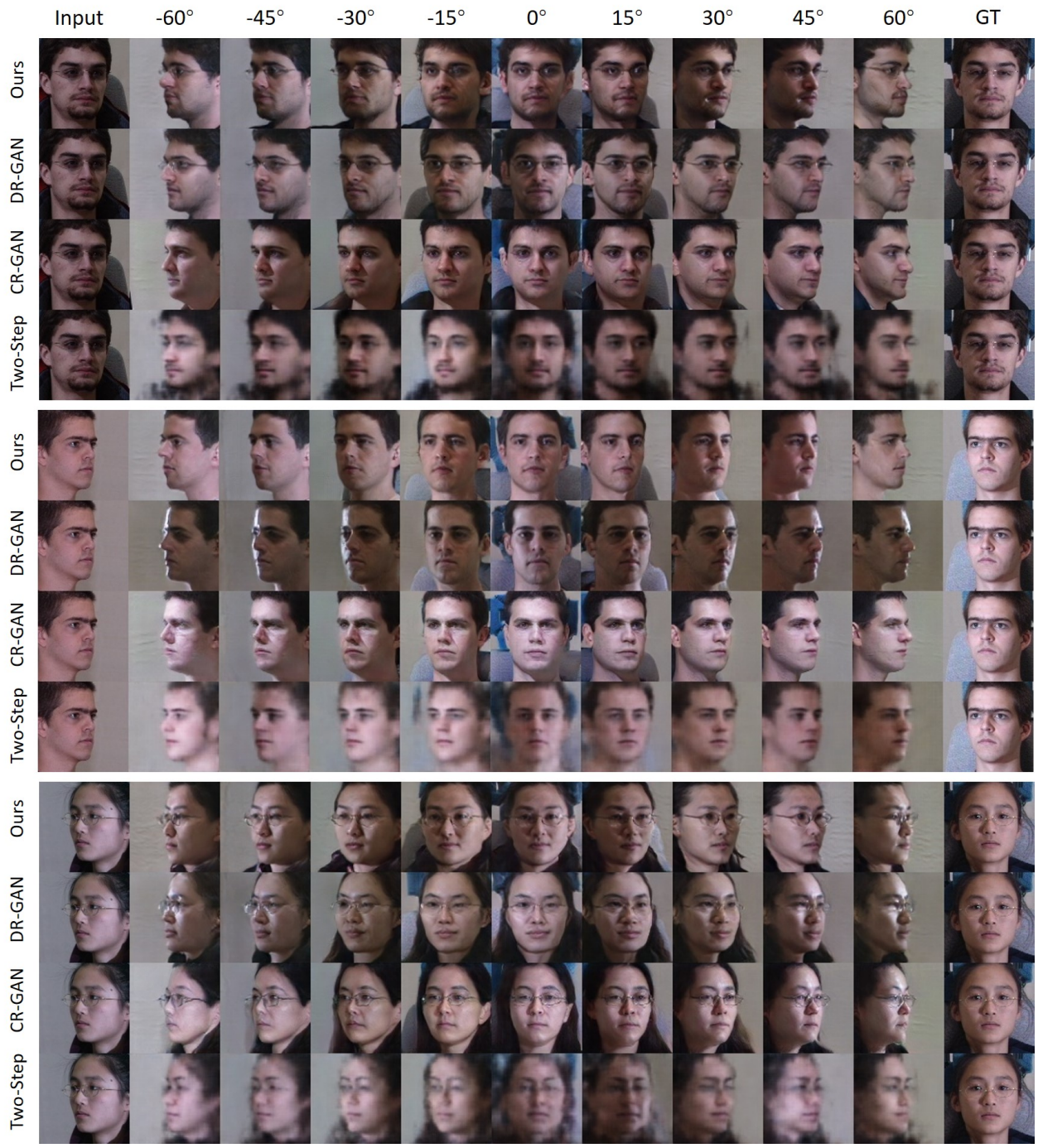


Figure 4. More results generated by different models on Multi-PIE dataset. From left to the right: input image, generated images with 9 different viewpoints and a frontal ground-truth face of the input face. Please pay attention to the overall identity of the generated faces as well as specific details such as jaw shape, hair, mouth and moustache which are important to identity preservation.

References

[1] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3, 2013. 1

[2] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008. 3

[3] Luan Tran, Xi Yin, and Xiaoming Liu. Disentangled representation learning gan for pose-invariant face recognition. In *CVPR*, pages 1415–1424, 2017. 1