Interactive Avatar Image Manipulation with Unconstrained Natural Language Instruction using Source Image Masking

Seitaro SHINAGAWA,[1,2] Koichiro YOSHINO,[1,3] Sakriani SAKTI,[1,2] Yu SUZUKI,[1]
Satoshi NAKAMURA[1,2]

Abstract

We propose an interactive image manipulation system with natural language instruction, which can generate a target image from a source image and an instruction sentence that describes the difference between the source and the target image. The system makes it possible to modify a generated image interactively and make natural language conditioned image generation more controllable. We construct a neural network that handles image vectors in latent space to transform the source vector to the target vector by using the vector of instruction. Additionally, we propose Source Image Masking (SIM), a method for masking source image to clarify the changing points of the generated image from the instruction. The experimental results indicate that the proposed framework successfully generates the target image using a source image and instruction on manipulation in our dataset. Moreover, introducing SIM makes the training more stable and faster.

1. Introduction

Interactions between an image generation system and a user with natural language are becoming popular. These interaction systems of an automatic image generation from a natural language description, often called as caption2image or cap2image, have been successful in generating image of flowers [5], birds [14] and various objects in MSCOCO [4], using Generative Adversarial Networks (GANs) [7, 8, 10] or Auto-regression models [12, 13].

Cap2image can be seen as a fundamental step toward various computer-aided design applications; however, there is a gap from practical use. One of the major problems of existing such systems is that users cannot modify a part of generated images, even if they are not satisfied with the details of generated images. If users want to modify any parts of generated images, they need to give new and more detailed descriptions. However, such re-generation often cause changes on not only the intended point but also some unintentional points in images. To migrate the difficulties, complementary inputs help to generate the desired image. Sharma et al. [10] used dialogue history data as complementary input to generate MSCOCO images. However, such a large text input is not intuitive for users of image modification.

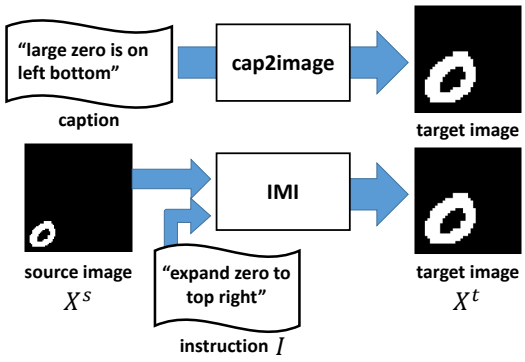

Fig. 1 Comparison of natural language conditioned image generation framework between cap2image (left) and image manipulation with instruction (right).

Whereas these cap2image approaches use only text information to generate an image, Image Manipulation with Instruction (IMI) [11] approach uses not only text information but also a source image as complementary input to generate a new target image. It also can be seen as an image-to-image translation problem conditioned by natural language. Figure 1 describes the difference between cap2image and IMI. Natural language instructions do not denote the descriptions of the target images, but the dif-

[1] Nara Institute of Science and Technology
[2] RIKEN, Center for Advanced Intelligence Project AIP
[3] PRESTO, Japan Science and Technology Agency
{shiangawa.seitaro.si8,koichiro,ssakti,ysuzuki,s-nakamura}@is.naist.jp

ferences between the source and the target images are described, i.e. "Expand zero to top right." IMI framework has advantages on not only providing a user a more intuitive and flexible modification interface but also making it easy to learn the generation system. The user does not need keeping or considering any previous input scripts about whole details of the image but only describing what part should be changed and how.

However, the prior work of IMI framework by extending cap2image suffers from co-occurrence of undesired changes on the generated image [11]. To solve this problem, we propose Source Image Masking (SIM) – masking a given source image to clarify the changing points of the generated image from the source image and given instruction.

2. Image manipulation with instruction (IMI)

As shown in Figure 1, image manipulation with instruction (IMI) is the task to generate the desired image X^t by fixing some parts of the given source image X^s. The change from the source image is given by a natural language instruction I, where $I = (w_1, w_2, \cdots, w_T)$. w_t denotes t-th word in the instruction I, and T represents the word length of I. During training, the model receives (X^s, X^t, I) triplets. In testing, the model outputs a generated image $\hat{X}^t$ from a given pair (X^s, I).

2.1 IMI system with extension from cap2image (baseline)

Deep Convolutional Generative Adversarial Networks (DCGAN) [6] are often used as a standard model for image generation with GANs. Reed et al. [8] succeeded to generate discriminative images from descriptions using DCGAN. Extending this model, Shinagawa et al. [11] provided an IMI model. Figure 2 shows the whole network architecture. W/ SIM denotes the proposed architecture in this paper, w/o SIM represents the architecture of Shinagawa et al. [11] as the baseline. The network is composed of four modules: *source image encoder* (ImEnc) with CNN [3], *instruction encoder* (IEnc) with 1-layer LSTM [1], *1-layer fully-connected layer*, and *image generator* (Gen and Dis) with DCGAN [6]. The source image feature ϕ^{im} and the instruction feature ϕ^i are extracted by using ImEnc and IEnc, as follows:

$$\phi^{im} = CNN_{ImEnc}(X^s) \quad (1)$$

$$\phi^i_t = LSTM_{IEnc}(w_t, \phi^i_{t-1}) \quad (\phi^i_0 = \mathbf{0}) \quad (2)$$

$$\phi^{fc} = FC(\phi^{im}, \phi^i_T) \quad (3)$$

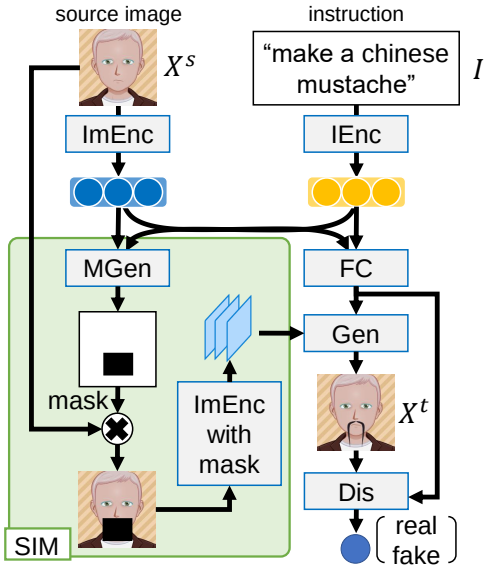

Fig. 2 Baseline (w/o SIM) and proposed (w/ SIM) network.

The output of the FC ϕ^{fc} represents the target image feature fed into *image generator*. For the training of *image generator*, matching aware method of Reed et al. [8] is used. The final output $\hat{y}$ represents the prediction of {real, fake}, which denote {match, mismatch} between the discriminator (Dis) output ϕ_d and the target image feature ϕ^{fc} fed into the generator of DCGAN (Gen).

$$\hat{X}^t = CNN_{Gen}(\phi^{fc}) \quad (4)$$

$$\phi^d = CNN_{Dis}(X) \quad (X \in \{X^t, \hat{X}^t\}) \quad (5)$$

$$\hat{y} = FC_{Dis}(\phi^d, \phi^{fc}) \quad (6)$$

This model demonstrated some successful changes in IMI task. However, the model sometimes modifies not only the part mentioned by the instruction but also the parts not mentioned.

2.2 The proposed source image masking (SIM)

The existing architecture of IMI system has a problem with changes which are not mentioned in the instruction. We propose source image masking architecture to suppress such undesired changes. There are two additional modules – *mask generator* (MGen) and *image encoder with mask* (ImEnc with mask) as follows:

$$mask = CNN_{MGen}(\phi^{im}, \phi^i_T) \quad (7)$$

$$\phi^{imm} = CNN_{ImEncwithmask}(X^s \odot mask) \quad (8)$$

where the mask (channel size: 1) in (7) is replicated to be the same channel to X^s (channel size: 3) in (8). MGen generates the mask which has [0,1] range. The mask and the source image are multiplied and fed into *ImEnc with mask*. Therefore, the mask works as a pixel-wise gate of the source image. We expect that ImEnc with mask extracts what part of the source image should be retained or changed.

3. Experimental settings

3.1 Avatar Image Manipulation with Instruction (AIMI) Dataset

We constructed a dataset for IMI task, which consists of triplets: (source image, target image, instruction). We collected avatar image pairs in AvatarMaker.com[1]. An avatar has various attributes, i.e., gender, eyes, mouth, background, etc. The construction procedure of the dataset is as follows:

(1) An source image is generated with random attributes.
(2) One attribute of the source image is changed to generate a target image.
(3) Both of the source image and the target image are shown to crowd workers on crowdsourcing, and they add an instruction that describes the difference between the source and target image.

After cleaning of these annotated instructions, we finally got 4,756 valid triplets of (source image, target image, instruction). We divided the data into train, validation, and test set in proportions of 90%, 5% and 5%, namely 4,296, 230, and 230 examples. We also collected randomly generated images without any instructions – total size: 161,065.

3.2 Training detail

During training, we repeated the training without instructions (w/o instruction) and with instructions (w/ instruction) phase of 2,200 examples alternatively.

In w/o instruction training phase, we utilize an image as a source and a target image. We train a model as an auto-encoder to generate the same image to a given source image. The instruction vector was set by zero vector. The aim of the w/o instruction training is supporting the *image-generator* learning to generate clear target images. Note that the masking layer is not trained in this phase because there are no differences between the source and target images.

In w/ instruction training phase, we utilized full triplets of (source image, target image, instruction). In SIM model, the ground truth of mask can be provided by comparing a pair of source and target image. Thus, we provide the ground truth mask, whose pixels are set by zero where the pixels in the same position of the source and target image are different, otherwise set by one. We also provide a mask loss function, mean squared error between a generated and the ground truth mask, to encourage SIM model to work correctly.

[1] http://avatarmaker.com/

Fig. 3 generated examples on each epoch.

We trained the models using *Adam* [2] ($\alpha = 2.0 \times 10^{-4}, \beta = 0.5$) until 5000 phases. Images are resized to 64×64. Hidden size is 128 for ϕ^i and ϕ^{fc}; 1024 for ϕ^{im}; $512 \times 4 \times 4$ for ϕ^{imm}. Batch size is 64. Vocabulary size is 1892. We used feature matching [9] to stabilize training.

4. Experimental results

Figure 3 indicates generated examples on each phase for an example in validation set. While w/o SIM took over 4,000 phases to generate a similar image to the target image, w/ SIM generated a more similar image under 1,000 phases. This indicates that introducing SIM makes the training more stable and faster. To select models for the evaluation, we looked at the loss curve line of feature matching in training. Although the training GANs is not stable, we found that feature-matching loss is relatively useful to select the better model. Judging from the generation results on each phase and the whole validation loss, we selected the model of phase 4,350 for the baseline and phase 700 for the proposed model, respectively.

We used test set (230 examples) for the quantitative evaluations. As the objective evaluation, we compared MSE scores between generated images and target images by using the baseline and the proposed model. The results were 4.92×10^{-2} for the w/o SIM (baseline) and 3.31×10^{-2} for the w/ SIM (proposed). It indicates that the generated examples by using the proposed model are better than that of baseline.

As the subjective evaluation, we showed a source image, an instruction, and generated images by using the baseline and the proposed model to human evaluators. We used crowd sourcing[2] for the evaluation. Each crowd worker provided a preference for randomly-ordered each pair of generated images (A,B) in five-grade (1: A is much better than B, 2: A is better than B, 3: the results are comparable, 4: B is better than A, 5: B is much better than the A). Considering reversed order cases, 460 examples were evaluated by human evaluators in total. Each example was evaluated by three workers and each worker evaluated up to 10 examples. Finally, we obtained 1,380 evaluated results. After the restoring of reversed-ordered

[2] Crowdflower: https://www.crowdflower.com/

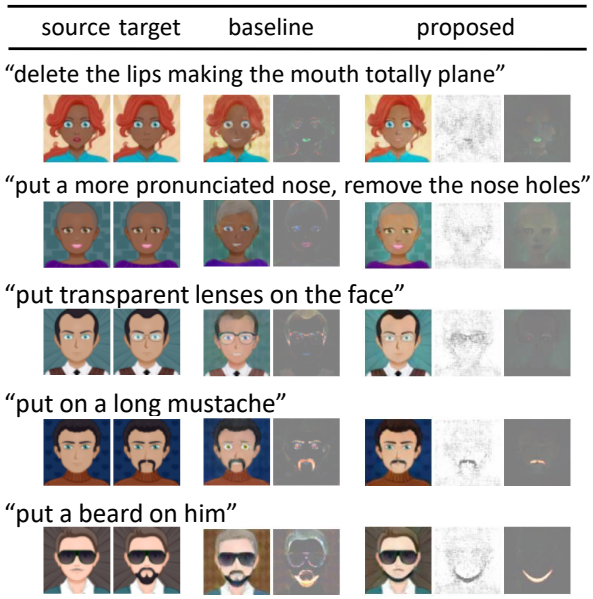

Fig. 4 The generated examples of each model.

examples, the proportion of each grade to the total is (1: 8.8%,2: 14%,3: 13%,4: 37%,5: 27%) where the workers who preferred the baseline selected 1 or 2, the workers who preferred the proposed selected 4 or 5. The total proportion of subjects who preferred the generated images of the proposed model (evaluated 4 or 5) is over 60%. It indicates that the proposed model generated better images than the baseline.

We compared generated examples using test set as qualitative evaluation. Figure 4 shows the generated examples using the baseline and the proposed model. The texts above each image line show the given instructions. From the left of each images line, the source image, the target image, generation results of the baseline method, generation results of the proposed method and the instruction are listed. The generation results of the baseline method have two images: the generated image and the visualization of changing points by the pixel-wise squared error between the source and the generated image. The generation results of the proposed method have three images: the generated image, the visualization of the mask and the visualization of changing points as well.

Both models successfully generated changed images according to the instructions; however, the baseline model suffers from the co-occurrence of undesired changes, i.e., the texture or color of the background, shape of hair, eyes or mouth. On the other hand, the proposed model successfully kept the details of the source images, which is not mentioned in instructions. We found that the proposed model can preserve details of source images; therefore, the model works better for small changing, i.e., changing nose, mouth, ears, which are difficult for the baseline model.

5. Conclusion

In this paper, we tackled the image manipulation with instruction (IMI) problem, by using natural language instructions. We proposed source image masking (SIM) to suppress undesired changes in the generated images. The generated examples of the proposed model had advantages on both of objective evaluation: mean squared errors to the reference, and subjective evaluation: human evaluation via crowdsourcing. Dealing with more complicated changing is a future work toward useful computer-aided design systems that can interact with users.

References

[1] Hochreiter, S. and Schmidhuber, J.: Long short-term memory, *Neural computation*, Vol. 9, No. 8, pp. 1735–1780 (1997).

[2] Kingma, D. and Ba, J.: Adam: A method for stochastic optimization, *In Proc. ICLR* (2015).

[3] LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. and Jackel, L. D.: Backpropagation applied to handwritten zip code recognition, *Neural computation*, Vol. 1, No. 4, pp. 541–551 (1989).

[4] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P. and Zitnick, C. L.: Microsoft coco: Common objects in context, *European conference on computer vision*, Springer, pp. 740–755 (2014).

[5] Nilsback, M.-E. and Zisserman, A.: Automated flower classification over a large number of classes, *Computer Vision, Graphics & Image Processing, 2008. ICVGIP'08. Sixth Indian Conference on*, IEEE, pp. 722–729 (2008).

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

[7] Reed, S., Akata, Z., Mohan, S., Tenka, S., Schiele, B. and Lee, H.: Learning What and Where to Draw, *In Proc. NIPS* (2016).

[8] Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B. and Lee, H.: Generative Adversarial Text-to-Image Synthesis, *In Proc. ICML* (2016).

[9] Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A. and Chen, X.: Improved techniques for training gans, *In Proc. NIPS*, pp. 2226–2234 (2016).

[10] Sharma, S., Suhubdy, D., Michalski, V., Kahou, S. E. and Bengio, Y.: ChatPainter: Improving Text to Image Generation using Dialogue, *arXiv preprint arXiv:1802.08216* (2018).

[11] Shinagawa, S., Yoshino, K., Sakti, S., Suzuki, Y. and Nakamura, S.: Interactive image manipulation with natural language instruction commands, *In Proc. NIPS-ViGIL* (2018).

[12] van den Oord, A., Kalchbrenner, N., Espeholt, L., Vinyals, O., Graves, A. et al.: Conditional image generation with pixelcnn decoders, *In Proc. NIPS*, pp. 4790–4798 (2016).

[13] van den Oord, A., Kalchbrenner, N. and Kavukcuoglu, K.: Pixel recurrent neural networks, *In Proc. ICML*, pp. 1747–1756 (2016).

[14] Wah, C., Branson, S., Welinder, P., Perona, P. and Belongie, S.: The Caltech-UCSD Birds-200-2011 Dataset, Technical report (2011).