PIVTONS: Pose Invariant Virtual Try-on Shoe with Conditional Image Completion

Chao-Te Chou, Cheng-Han Lee, Kaipeng Zhang,
Hu-Cheng Lee, Winston H. Hsu

National Taiwan University, Taipei, Taiwan

Abstract. Virtual try-on – synthesizing an almost-realistic image for dressing a target fashion item provided the source human photo – has growing needs due to the prevalence of e-commerce and the development of deep learning technologies. However, existing deep learning virtual try-on methods focus on the clothing replacement due to the lack of dataset and cope with flat body segments with frontal poses providing the front view of the target fashion item. In this paper, we present pose invariant virtual try-on shoe (PIVTONS) to cope with the virtual try-on shoe. We collect the first paired feet and shoe virtual-try on dataset, Zalando-shoes, containing 14,062 shoes among the 11 categories of shoes. The shoe image only contains a single view of the shoes but the try-on result should show other views of the shoes depending on the original feet pose. We formulate that as an automatic and labor-free image completion task and design an end-to-end neural networks composing of feature point detector. By combing three losses for image generation, we can synthesize realistic results. Through the numerous experiments and ablation studies, we demonstrate the performance of the proposed framework and investigate the parameterizing factors for optimizing the challenging problem.

Keywords: Virtual try-on, Generative model

1 Introduction

In recent years, the demands of online shopping for fashion items are increasing, and are predicted to increase in the future. When shopping online for fashion items, consumers will consider what they look like in those fashion items. Therefore, virtual try-on systems for different fashion items are in need. They can assist consumers more effectively and precisely to find the fashion items they want and reduce the risk for the retailer to have an additional cost if the consumers want a return.

To use a virtual try-on system, a user should provide a *source image*, which is an image of a person or body parts intending to try a different fashion item, and an image of the target item, a virtual try-on system synthesizes a photo-realistic new image, a *target image*, by overlaying the given target fashion item seamlessly onto the corresponding region of body parts in the source image. Some examples

Fig. 1: **Results generated by our PIVITONS.** For our virtual-try on shoes, the inputs are a source image with feet wearing shoes and a target item, a single view of a single shoe. We can see that the poses of feet are diverse and only a single view of the target shoe is given. With this strict constraint, our virtual try-on method swaps the shoes onto the feet in the source image and generate target image meeting the criteria for a virtual try-on system.

are shown in Fig. 1. The generalized criteria for virtual try-on should include : (I) The fashion item in the target image should be the same as the target fashion item; (II)The region outside the fashion item in the target image should be the same as that of the source image; (III) The target image should be realistic and consistent. Existing works using deep learning for virtual try-on have been focusing on virtual try-on clothing. CAGN [7] introduces a cycle loss to replace the clothing texture. VITON [5] utilizes human parsing to guide the synthesis of target clothing onto human body. Both method designed for virtual try-on clothing with frontal-view human body providing frontal view of clothing. Other virtual try-on systems have not been developed due to the lack of dataset. In this paper, we present a pose invariant virtual try-on shoe (PIVITONS) method with conditional image completion to address the virtual try-on shoe [1] problem. We collect the dataset, Zalando-shoes, providing feet-shoe pairs. Virtual try-on shoe is challenging because the poses of human feet are diverse and the shoe, target fashion item, only show the single view of a single shoe, which is a pose difference between the source image and the target shoe. Thus, we call our method pose invariant. To formulate our method into a conditional image completion problem, we first detect the bounding box of shoes in the source image. We hide the region inside bounding and generate the target image considering the region outside bounding box and the conditioned target shoe.

To address the problem of diverse poses and pose information missed inside the bounding box, we use a key-point detector to detect the four key-points, right

[1] A video is uploaded in the supplementary materials.

toe, right heel, left toe, and left heel, of shoes in the source image. The detected key-points assist the synthesis of target shoes to the correct position and size onto feet in the source image. Therefore, we only need to annotate six points, the two diagonal points of the bounding box and four key-points for the shoe, for our ground truth data. Furthermore, We combine $l2$ loss, perceptual loss [8], and adversarial loss [4] to realistically synthesize target image that can not only show the shown part of shoe shown in the target image but also show the missing part. Thus, the pose difference between target shoe and source image could be solved.
Our contributions can be summarized as following:

1. To the best of our knowledge, we are the first to tackle the pose invariant virtual try-on shoe problem with the combination of detection and conditional image completion.
2. We collect the first Zalando-shoes dataset providing feet-shoe pairs for virtual try-on shoe and provide ground truth bounding box and key-points of shoes in source images. The dataset will be made public in the future for further research and comparison.
3. The proposed method can generate perceptually realistic images, which is a strong baseline for Zalando-shoes dataset in the task of virtual try-on shoe.

2 Related Work

Generative adversarial network. GAN [4] has been one of the most widely used deep generative models. It demonstrates promising results in image generation [17]. The generated image can be controlled by combining GAN with other conditional priors such as text [18, 23], or discrete labels [14]. Some other works for image-to-image translation use $l2$ or $l1$ loss with GAN to force the output to be conditioned on the input image. Pix2pix [6] uses GAN in a conditional manner with $l1$ loss and proposed a general framework for many tasks of image-to-image translation such as image colorization and image completion to name a few. The presented patchGAN is also used in our work. SRGAN [10] combines per-pixel loss, perceptual loss [8] and GAN for the task of super-resolution. In our work, we also apply these losses to generate realistic results for our task of virtaul try-on shoe with conditional image completion. FashionGAN [25] edit a fashion item on person wear according to the given text description by first generating the human body segmentation and then synthesizing a new fashion item corresponding to the text. It can be regarded as text-based virtual try-on. PG^2 [13] design a two stages network to generate person image of different view with the providing human image and key-points of target pose. In our work, we also provided four key-points of shoes in the source image. However, the function of key-points in our work is to provide pose information inside the bounding box.

Image completion. Traditional diffusion approach [1] utilizes a diffusion equation to smoothly propagate information into the missing region from its sur-

rounding area. The inpainting region is sharp and without color artifacts. Recently, deep learning and GAN-based approaches have been utilized and show decent results. Rolf *et al.* [9] exploited the shape information of the missing regions and train a neuron network for denoising and inpainting of small regions. Context encoder [16] proposed an unsupervised visual feature learning algorithm driven by context-based pixel prediction. Yu *et al.* [22] proposed a fully convolutional neural network which utilizes surrounding image features as references during network training to make better predictions to fill missing regions in an image. Li *et al.* [11] proposed an image completion algorithm using a deep generative based on GAN [4] by introducing a global and a local discriminator as adversarial losses and a semantic parsing network to provide face parsing loss. It can generate missing facial key parts from random noise and show plausible visual results. Different from these image completion tasks, our task not only need to fill in the missing part with the surrounding information from the rest of the image and the learned knowledge from the dataset but also need to take target shoe as a condition into consideration to generate visually plausible results.

Virtual try-on. VITON [5] is an image-based virtual try-on method utilizes only 2D information. Given a human try-on image and the image of target fashion item, it transfers the target fashion item onto the corresponding region of a person. It utilizes human segmentation maps to guide the generation of target fashion item onto the human body. With the state-of-the-art person pose detector [2] and human parser [3], this method can get good quality segmentation and key-points of the human body. CAGAN [7] use a cycle loss [24] to train a virtual try-on network. Therefore, it does not need segmentation or fashion item detection to swap a fashion item onto a human body. However, during training and testing, it needs the product images of both the target item and the original item in the human try-on image. This requirement is not convenient for users to try-on new fashion item with their own source image. Different from existing works, our method is designed for the virtual try-on shoe, which is the best of our knowledge has not been solved before. Our method uses the detection of a bounding box and key-points in a person try-on image to dress the person in the image with a new fashion item. Furthermore, our method does not need the image of the original fashion item the person wear to swap new fashion item onto that human body, which can make it as feasible for real-world applications as VITON [5].

3 Proposed Method

In this section, we present our PIVTONS for the virtual try-on shoe. Given a source image I with feet wearing shoes, we aim to generate target image $\hat{I}$ with the feet in the source image wearing the target shoe I_s. To incorporate supervised learning, a straightforward approach is to collect training data of a person with fixed pose wearing different shoes and the corresponding target shoe images. However, it is expensive and time-consuming to collect this kind of data.

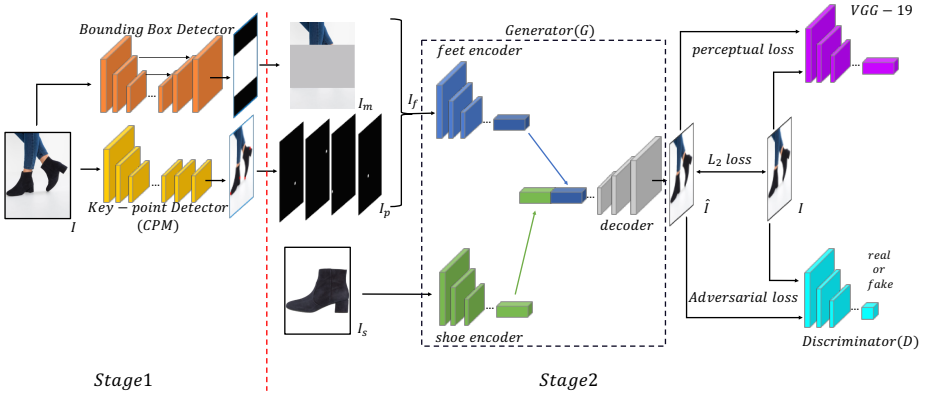

Fig. 2: **Proposed Pipeline.** Our pipeline composes of 2 stages. The first stage is composed of a bounding box detector and a key-point detector. Those networks are used to predict bounding box and key-points in the source image to prepare the masked source image I_m and the key-point maps I_p for the input of the next stage. The second stage is the main network of our conditional image completion virtual try-on network. It is used to filling the missing part to generate the target image. Please refer to sec. 3 for more details.

Therefore, we use the source image as ground truth during training and both the source image and target shoe are the inputs of our pipeline.

However, directly giving the network the original source image I without any preprocessing along with a target shoe I_s and using the same source image as ground truth to train the network would cause the network to output the source image I directly during testing despite a different shoe is provided. Therefore, we formulate our problem as an image completion problem. We detect the bounding box and key-points (sec. 3.1) of shoes in source image, hide the part inside the bounding box, and synthesize the full target image $\hat{I}$ (sec. 3.2). With the region outside the bounding box and the target provided to our pipeline, our method could synthesize the target image meeting the first and second virtual criteria described in sec. 1. To make the generated target images consistent and realisic to meet the third criteria of virtual try-on described in described in sec. 1, we combine the per-pixel loss, perceptual loss [8] and the adversarial loss [4] (sec. 3.3). The proposed pipeline with the networks is shown in Fig. 2.

3.1 Stage-1 : Detectors

Bounding Box Detector. For the task of virtual try-on, there is often a single person and a single fashion item to be swapped in a source image, so only a single bounding box is needed in our task. Therefore, we design a simple network for our task of bounding box detection. During training, we generate a binary mask by filling the region inside bounding box with ones and outside with zeros with

respect to the ground truth bounding box. We then train a convolutional network for semantic segmentation with only two classes, foreground and background. During testing, our network predicts a mask indicating where the foreground is and then we can get the bounding box for the part of shoes in the source image through the boundary of that region. To provide more information about the target shoe that will be generated in this hiding region in the later stage, we fill the region inside the bounding box with the mean color of the shoe image I_s. We call the resulting image as a masked source image I_m.

Key-point Detector. Although I_m can provide the information about the pose the network need to generate, sometimes it can not provided clear information. To preserve the feet pose inside the bounding box after swapping the shoe and generate more precise shoe size, we use key-points to help the generation of the target image. In our work, we use CPM [21] with six stages and use the belief map generated in the last stage as the final key-points detection result. After getting the belief maps for each key-point, we take the point with the maximum value as the center of the key-point. To leverage spatial layout, each key-point is then transformed into a Gaussian peak heatmap and the value at (x, y) on the heatmap for a key-point is given by the following equation.

$$f(x, y) = exp(-\frac{(x - x_0)^2 + (y - y_0)^2}{\sigma^2}) \tag{1}$$

where (x_0, y_0) is the position of the key-point. The four individual heatmaps are further stacked into 4-channel heatmaps I_p.

3.2 Stage-2 : Conditional Image Completion

Given the masked source image I_m, key-point heatmaps I_p, and the image of the target shoe I_s, the network in stage-2 directly generates the full target image. The stage-2 network of our image completion virtual try-on consists of two encoders and one decoder. The architectures of both encoders are the same, but the parameters are not shared. For simplicity, one encoder is called the feet encoder, the other is called the shoe encoder. The input of feet encoder is I_f, which is the stack of I_m and I_p. This encoder extracts the feature and provides the information about the feature of pant and the pose of the feet. The input of the shoe branch is the target shoe image I_s. It can be seen as a conditional branch to provide information about the shoe to try-on. It extracts the feature and provides the information about the texture, shape, and color of the target shoe. The features from feet encoder and shoe encoder are concatenated and fed into the decoder to generate the predicted image $\hat{I} = G(I_f, I_s)$.

3.3 Loss Function

Per-pixel loss. For image generation task with ground truth image, a simple way to train the network is using a per-pixel $l1$ or $l2$ loss. In our work, we choose

to use $l2$ loss and its formula can be written as

$$L_{l2}(G) = \frac{1}{H \times W \times 3} ||\hat{I} - I||_2^2 \tag{2}$$

H and W are the height and width of the input three channels image, respectively.

Perceptual loss. Since $l2$ loss generates smooth and blurred images, we add perceptual loss [8], which minimizes the difference between corresponding feature maps of the synthesized image and the ground truth image, computed by a pre-trained CNN network. The formula for perceptual loss can be written as

$$L_{perc}(G) = \sum_{i=1}^{i=5} \lambda_i ||\phi_i(\hat{I}) - \phi_i(I)||_2^2 \tag{3}$$

ϕ is a VGG-19 network pre-trained on ImageNet for image classification. Therefore, $\phi_i(x)$ is the feature map of image x of the i-th layer in the network ϕ with shape $H_i \times W_i \times C_i$. We utilize 'conv1-2', 'conv2-2', 'conv3-2', 'conv4-2', 'conv5-2' of the VGG-19 model. Following perceptual loss [8], We choose $\lambda_i = \frac{1}{H_i \times W_i \times C_i}$.

Adversarial loss. To make our generated images sharper and more realistic, the adversarial loss is added. In GAN [4], the generator is trained to produce realistic images and aims to deceive the discriminator to classify them as the real image. Therefore, the loss for the generator can be written as

$$L_{adv}(G) = log(1 - D(\hat{I})) \tag{4}$$

The discriminator is trained to distinguish between real images in the dataset and fake images generated by the generator. It would guide the generator to learn to generate realistic images similar to images in the distribution of real data. The loss for the discriminator can be written as

$$L_{adv}(D) = -[log(1 - D(\hat{I})) + log(D(I))] \tag{5}$$

Overall, our loss for the generator can be written as

$$L(G) = L_{l2}(G) + \alpha \cdot L_{perc}(G) + \beta \cdot L_{adv}(G) \tag{6}$$

α and β are constants controlling the ratio between these three losses. By simultaneous training generator and discriminator, the generator can learn to generate images in the real image distribution.

4 Experiment

In this section, we describe the details of how we collect our Zalando-shoes dataset (sec. 4.1), implementation of our method (sec. 4.2). We construct a

Fig. 3: **Example feet-shoe pairs in our Zalando-shoes dataset.** The first row shows the human try-on feet images with the ground truth bounding box and key-points. The second row is the corresponding shoes in the source image. In these example, we can see that the feet poses are diverse and complicated. The shoe types also vary a lot with different shapes and sizes. More examples are shown in the supplementary material.

baseline method and compare with it (sec. 4.3) because we are the first to tackle virtual try-on shoe. We also conduct three analysis (sec. 4.4) to demonstrate the effectiveness of the design of our method. Since the task of virtual try-on is to try on a fashion item different from the original one in the source image, existing evaluation methods such as SSIM [20] and PSNR for image completion are not suitable for evaluating our experiment results owing to the lack of ground truth. The inception score (IS) [19] is usually used to evaluate the quality of the generated images synthesized by the generative models. However, it tends to reward sharper image content generated by adversarial training and has shown failed to faithfully evaluate the generated target images of virtually try-on in VITON [5]. Therefore, in each separate experiments, we show some examples to qualitatively evaluate our method. All the examples shown in the experiments are from the test pairs described in sec. 4.1. We draw the predicted bounding box and key-points in the source images shown in the following experiments. To quantitatively evaluate our method, a user study is conducted (sec. 4.5). In the end, we show the situation in which our method may fail in sec. 4.6.

4.1 Zalando-shoes Dataset

Existing dataset [12] provides images with human wearing shoes and shoes. However, the region of shoes in the human try-on image is small. Furthermore, it does not provide human-shoe pairs. Therefore, we collected our dataset , Zalando-shoes, from the website of Zalando[2]. We first crawled all the shoe and human try-on image pairs from the website. All images were then cropped and resized

[2] https://www.zalando.co.uk

Table 1: **Number of feet-shoe pairs in different categories in our dataset.** In the table, we show the exact number of feet-shoe pairs in the different categories of our dataset. There are total eleven categories in our zalando-shoe dataset. Furthermore, shoes in different categories have different shape and feature. More examples of image pairs are shown in the supplementary material.

Category	boots	flats-lace-ups	sandals	slippers	ballet-pumps	trainers
# pairs(train/test)	521/107	1197/211	1634/285	156/32	309/38	1959/389

Category	sports-shoes	heels	ankle-boots	mules-clogs	flip-flops-beach-shoes
# pairs(train/test)	97/15	1639/344	3794/697	524/59	47/8

to 256×192. We manually removed image pairs including some part of shoes outside the human try-on images and annotated the remaining 14,062 pairs with bounding box and key-points. The bounding box is annotated such that the full pair of shoes is inside the bounding box and both ankles of human feet are inside bounding box. Some examples are shown in Fig. 3. Those remaining images were further randomly split into train set and test set with 11,877 and 2,185 pairs respectively. We have shown the details of the number of pairs for different categories in train and test set in Table. 1. Note that we split the same type of shoe with different colors into the same set. To test the feet to try on shoes different from the original ones, we randomly shuffled the shoe product images 5 times in this 2,185 test pairs for evaluation, and all our results shown in the experiments are from these randomly shuffled pairs. The code, dataset, and the shuffled feet-shoe pairs for testing will be released for further research and comparison in the future.

4.2 Implementation Details

Network architecture. The encoders in stage-2 are composed of a series of convolution layers with a kernel size of 5 and a stride of 2, and their numbers of filters are 64, 128, 256, 512, 1024, respectively. After the feature maps extracted by feet encoder and shoe encoder are concatenated, the concatenated 2048 channels feature maps then undergoes a convolution layer with a kernel size of 1, a stride of 1, and 1024 filters to perform dimension reduction. Transpose convolution layer tends to introduce checkboard artifact [15], so our decoder consists of a series of bilinear interpolation 2× upsample. Each 2× bilinear upsampling is followed by a convolution layer with a kernel size of 3 and a stride of 1, whose number of channels are 1024, 512, 256, 128, 64, respertively. All neurons use the *ReLU* as the activation function. In the end, a convolutional layer with a kernel size of 1, a stride of 1 and 3 filters with the *tanh* as the activation function is applied to the output of the encoder to generate a $256 \times 192 \times 3$ image. We use the same discriminator as that of patchGAN [6] and the setting of optimizer for training the discriminator is the same as the generator.

Training step. We use the RMSProp optimizer with decay = 0.9, epsilon = $1e-10$ with an initial learning rate of 1e-4. We choose $\sigma = 3$ for the Gaussian peak in eq.(1). For the generator loss in eq. 6, we choose $\alpha = 1$ and $\beta = 0.01$. We train the network for 50 epochs with a batch size of 16. We use the ground truth bounding box and key-points to train our stage-2 network.

4.3 Compared Approach and Result

We compare our method with a modification of pix2pix [6] called pix2pix-m. The input of the generator is the concatenation of I_m, I_s and I_p. The input real data to the discriminator is the concatenation of I, I_s and I_p. We first modified the number of layers of the original network to fit image size of 256×192. We also add one convolutional layer with filter size 3 and stride 1 after each transpose convolutional layer and convolutional layers for the generator. The number of channels for each added convolutional layer is the same as the convolutional or transpose convolutional layers before them. The *lamda* of the loss is set to 0.001. The results are shown in Fig. 4.

4.4 Components Analysis

In this section, we conduct three analysis to demonstrate the effectiveness of our designed pipeline. We first show the results of the combination of different losses and conclude that we use the combination of losses in an effective manner in sec. 4.4.1. Second, we demonstrate that the key-points are import components for our method in sec. 4.4.2. Third, we verify that using the source image I to replace I_m in the second stage of our pipeline will cause the network to output the source image during testing and ignore the different target shoes in sec. 4.4.3. Therefore, our design of hiding the shoes part in the source image is an import component in our pipeline.

4.4.1 Analysis of different combination of losses Different losses have different properties, and will result in different target images, so we show some examples in Fig. 5 to demonstrate the combination of losses we use is effective. Our method using only $l2$ loss generates smooth and blurred results. Furthermore, when the target shoe is white shown in column (f), $l2$ fails to generate a clear edge for the shoes in the target image. By adding perceptual loss [8], the target images show more details of shoes and pants, which meets the feature provided in the target shoe and source image respectively, and it can clearly show the edge of white shoes in column (f). After adding the adversarial loss, the results image become more realistic. It generates reflection in the target images shown in the column (a) and (c) in Fig. 5, and better texture. It also generates shoelace, which is not clearly shown in the image of target shoe, for the shoe in target image shown in column (b). Furthermore, in columns (d) and (e), we see that adding adversarial loss can eliminate some defect and produce better target images. It also correct the wrong color of paint in column (b). Thus, the combination of these losses is effective.

Fig. 4: **Compared with pix2pix-m.** The results show that our method performs favorably against the compared method. See more details in sec. 4.3

4.4.2 Analysis of key-points There are information losses about the feet pose in the cropped region. Therefore, key-points are used to provide pose information in the missing region. In Fig. 6, we show the results with and without key-points. In column (a), we show that the left shoe generated by the without key-points setting is shorter than that in the source image. Column (b), (c), and (d) shows that the generation of the direction and shape of shoes may fail without the guidance of key-points. Column (e) and (f) show that the setting of not using key-points may not be able to generate a full shoe.

4.4.3 Analysis of hiding the region of shoes in source image To demonstrate that the masked source image I_m is a valid component, we compare the generated target images of our original setting using masked source image I_m and the setting of replacing I_m with the source image I as input to our second stage. From the results show in Fig. 7, we validate that taking the original source image as input during training and testing will cause the network to ignore the target shoe when generating target image. In conclusion, hiding the part of shoes in the source image is an import technique for our method.

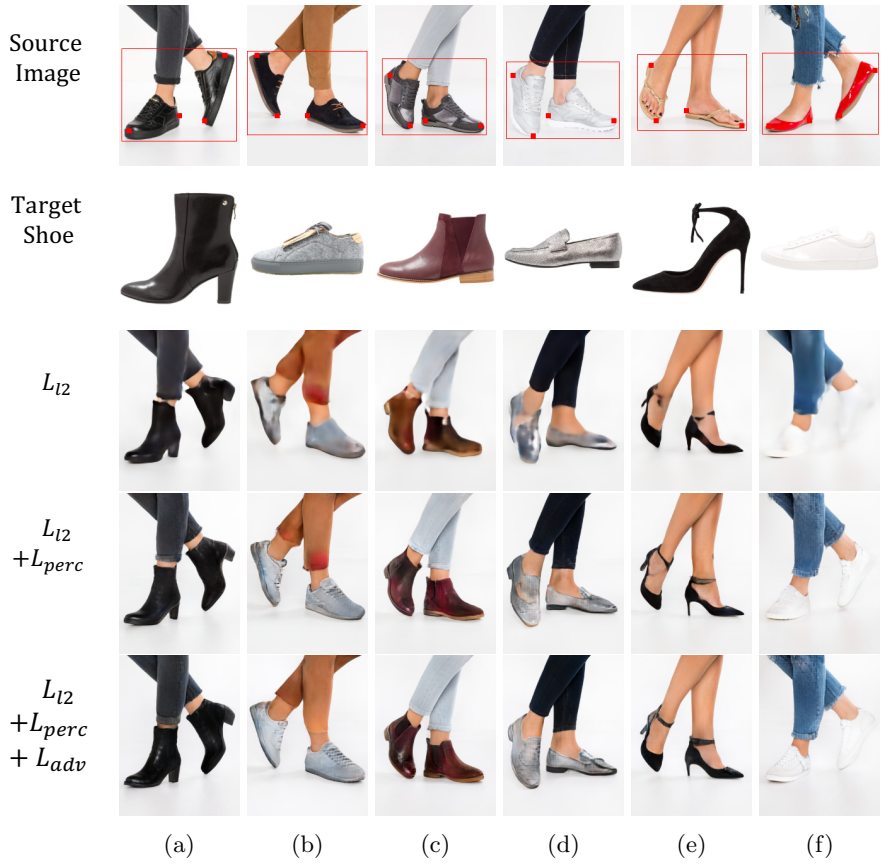

Fig. 5: **Analysis–the combination of losses.** The results show that using all three losses performs favorably against using only L_2 or $L_2 + L_{percep}$. See more details in sec. 4.4.1

4.5 User Study

A total of 20 volunteers participate in our user study. Two hundred try-on pairs are randomly selected from our test set and each pair is evaluated by 2 different workers. In each questions, the volunteers are provided with the source image, target item, and 5 target images generated by pix2pix-m and other different settings at the same time. They rank those images according to which image is more similar to their expecting target image. Two target image are not allowed to be ranked equally. We further compare the rank between our method and the pix2pix-m, or other settings to get the pairwise comparison results. The results are shown in Table. 2 . We show that our method with all three losses performs favorably against pix2pix-m and other settings.

Fig. 6: **Analysis–the importance of key-points.** We compare the results between with key-points and without key-points. The setting of without key-points does not use key-points during training and testing. From the results, we demonstrate that the information of key-points are important components. Please see sec. 4.4.2 for more details.

4.6 Failed cases

In Fig. 8, we show some failed cases. In column (a) and (b), our method sometimes fails to synthesize the target image properly when there are a big region of skin and toes needed to be generated. In column (c), the left feet is hard to generate properly because the left heel is not shown in the source image. Furthermore, our method fails to synthesize some kind of shoe onto non-proper feet pose as shown in column (d), (e), (f) because our method is designed to preserve the feet pose in source image when generating the target image. In column (d) and (e), when there are shoes with flat bottom in source image and the provided target shoe with heels, our method fails to generate a feasible target image because the pose of the feet in source image is not suitable to wear shoes with heels. In column (f), we show that swapping a shoe with flat bottom to the feet wearing shoes with heels will result in a awkward image.

Fig. 7: **Analysis–the effectiveness of the masked source image I_m.** In this figure, we compare the results of the generated target images between the original setting of using the masked source image I_m and using the source image I to replace I_m as input to our second stage. The results show that the network in the second stage will ignore the target shoe and generate image almost the same as the source image without using I_m for training. More discussions are shown in sec. 4.4.3.

5 Conclusion

In this paper, we deal with the problem of pose invariant virtual try-on shoes. We collect a dataset, Zalando-shoes and provide bounding box and key-points annotations to address the limitation of existing datasets for virtual try-on shoe. We formulate our problem as a conditional image completion problem and use key-points of feet in the source image to assist the synthesis of target shoes onto the correct position. In the experiments, the presented method shows promising qualitatively results meeting the criterion for virtual try-on. Although our method is designed for virtual try-on shoe in this paper, our future work is to extend the idea of conditional image completion to other fashion items.

Table 2: **User study–pairwise comparison.** Each cell represents the fraction of the number of target images generated by PIVITONS (all three losses and with key-points) are ranked better than one of the baseline or different analysis settings. Please refer to sec. 4.5 for more experimental details.

Method	pix2pix-m	L_2	w/o key-points	$L_2 + L_{perc}$
v.s. PIVITONS	0.8925	0.8825	0.745	0.55

Fig. 8: **Failed cases.** In this figure, we show some examples that our method fails to generate perceptually realistic results. Please refer to sec. 4.6 for more discussions.

6 Acknowledgement

This work was supported in part by the Ministry of Science and Technology, Taiwan, under Grant MOST 107-2634-F-002-007. We also benefit from the grants from NVIDIA and the NVIDIA DGX-1 AI Supercomputer. We also appreciate the research grants from Microsoft Research Asia.

References

1. Bertalmio, M., Sapiro, G., Caselles, V., Ballester, C.: Image inpainting. In: SIGGRAPH (2000)
2. Cao, Z., Simon, T., Wei, S.E., Sheikh, Y.: Realtime multi-person 2d pose estimation using part affinity fields. In: CVPR (2017)

3. Gong, K., Liang, X., Zhang, D., Shen, X., Lin, L.: Look into person: Self-supervised structure-sensitive learning and a new benchmark for human parsing. In: CVPR (2017)
4. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: NIPS (2014)
5. Han, X., Wu, Z., Wu, Z., Yu, R., Davis, L.S.: Viton: An image-based virtual try-on network. In: CVPR (2018)
6. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: CVPR (2017)
7. Jetchev, N., Bergmann, U.: The conditional analogy gan: Swapping fashion articles on people images. In: ICCV (2017)
8. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: ECCV (2016)
9. Köhler, R., Schuler, C., Schölkopf, B., Harmeling, S.: Mask-specific inpainting with deep neural networks. In: GCPR (2014)
10. Ledig, C., Theis, L., Huszar, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., Shi, W.: Photo-realistic single image super-resolution using a generative adversarial network. In: CVPR (2017)
11. Li, Y., Liu, S., Yang, J., Yang, M.H.: Generative face completion. In: CVPR (2017)
12. Liu, Z., Luo, P., Qiu, S., Wang, X., Tang, X.: Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In: CVPR (2016)
13. Ma, L., Jia, X., Sun, Q., Schiele, B., Tuytelaars, T., Van Gool, L.: Pose guided person image generation. In: Advances in Neural Information Processing Systems. pp. 405–415 (2017)
14. Mirza, M., Osindero, S.: Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784 (2014)
15. Odena, A., Dumoulin, V., Olah, C.: Deconvolution and checkerboard artifacts. Distill (2016). https://doi.org/10.23915/distill.00003, http://distill.pub/2016/deconv-checkerboard
16. Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A.A.: Context encoders: Feature learning by inpainting. In: CVPR (2016)
17. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434 (2015)
18. Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., Lee, H.: Generative adversarial text to image synthesis. arXiv preprint arXiv:1605.05396 (2016)
19. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training gans. In: NIPS (2016)
20. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. TIP **13**(4), 600–612 (2004)
21. Wei, S.E., Ramakrishna, V., Kanade, T., Sheikh, Y.: Convolutional pose machines. In: CVPR (2016)
22. Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.S.: Generative image inpainting with contextual attention. arXiv preprint arXiv:1801.07892 (2018)
23. Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X., Metaxas, D.N.: Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In: ICCV (2017)
24. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: ICCV (2017)
25. Zhu, S., Urtasun, R., Fidler, S., Lin, D., Change Loy, C.: Be your own prada: Fashion synthesis with structural coherence. In: ICCV (2017)