UNPAIRED IMAGE-TO-IMAGE SHAPE TRANSLATION ACROSS FASHION DATA

Kaili Wang[1] *Liqian Ma*[1] *Jose Oramas M.*[1 3] *Luc Van Gool*[1 2] *Tinne Tuytelaars*[1]

[1] *KU Leuven, ESAT-PSI* [2] *ETH/D-ITET/CVL* [3] *University of Antwerp, imec-IDLab*

ABSTRACT

We address the problem of unpaired geometric image-to-image translation. Rather than transferring the style of an image as a whole, our goal is to translate the geometry of an object while preserving its appearance. Our model is trained without the need for paired images. It performs all steps of the shape transfer within a single model and without additional post-processing stages. Experiments on clothing-based datasets show the effectiveness of the proposed method.

Index Terms— Image Generation, Image representation.

1. INTRODUCTION

Image-to-image translation (I2I) refers to the process of generating a novel image, which is similar to the original input image in some ways yet different in others. Typically, the input and output images belong to different *domains*, with images in the same domain sharing a common characteristic, e.g. going from photographs to paintings [14], from greyscale to color images [3], or from virtual (synthetic) to real images [37]. Apart from direct applications [17], I2I has proven valuable as a tool for data augmentation [4] or to learn a representation for cross-domain image retrieval [7].

Traditionally, each domain is characterized by a different appearance or *style*, and I2I is therefore sometimes referred to as *style transfer* [14]. While the translation process may drastically change the appearance or style compared to the input image, in many cases the image semantics are to be preserved, i.e. both input and output should represent the same objects and scene. Moreover, usually also the image geometry, i.e. the shape of the objects and the global image composition, is preserved. We refer to this as the image *content*. Most I2I methods build on top of Generative Adversarial Networks (GANs) [6, 28, 27, 2] to learn the translation. While some methods require paired data [12, 34, 36], some recent methods do not [28, 38]. To constrain complexity, the training data is often restricted to a specific setting, e.g. close-ups of faces [9, 36], people [25, 26], traffic scenes [22], etc.

In contrast to the traditional setting [26, 36], we focus on the case where input and output do *not* belong to domains that share the same geometrical information. Instead, we work with one object-centric domain with standard shape and one that is more contextualized with large shape variation (using a reference image to provide the right context). For instance, we go from a single piece of clothing to a person wearing that same item (see Fig. 1). This setting is significantly more challenging, as the image geometry changes. At the same time, the image semantics (e.g. the clothing pattern) should be preserved. Analogous to the term style transfer, we refer to this as *shape transfer*. While a couple of recent works [26, 34, 36] have looked into this setting, to the best of our knowledge, we are the first to propose a solution that does *not* require paired data, across different domains, for training.

Fig. 1. Translating a clothing item from a "catalog" image domain to a domain of individuals wearing the item (try-on task, top), and vice versa (take-off task, bottom). Notice how for both tasks the appearance details of the clothing items are preserved while their shape is effectively translated.

Our contributions are three-fold: i) We analyze the task of ***unpaired shape translation***. To the best of our knowledge, we are the first doing this in an unpaired manner. ii) We propose unpaired Shape Transformer, a method which does not require paired data or post-processing refinements. In one stream, an object with standard shape is transformed to a contextualized domain with arbitrary shape, and vice versa in the other stream. iii) We achieve a one-to-many mapping via context and structure information guidance.

2. RELATED WORK

In recent years I2I translation has received significant attention [12, 38, 1, 24, 11]. Most of these efforts focus on style/appearance transfer where the content depicted in the input and output images has an aligned geometric structure. [25, 26, 29, 9, 36] aim at the case when the geometry itself is to be transferred. However, these methods focus on translation between similar domains (e.g. person-to-person and face-to-face), with smaller variability compared to our setting (person-to-clothing)

[34] propose one of the first methods addressing cross-domain pixel-level translation. Their method semantically transfers a natural image depicting a person (source domain) to a clothing-item image corresponding to the clothing worn by that person on the upper body (target domain), and vice versa. Recently, [8, 32] propose two-stage warping-based methods aimed at virtual try-on of clothing items. These methods rely on paired data to learn to transfer the shape in a first stage and then refine it in a second stage. In contrast, we propose a more general method that utilizes the context and shape guidance to perform translation across different domains without any paired data. In addition, our method is able to handle the full appearance-preserving translation, in both directions, within a single model/stage.

Outside of the I2I literature, methods based on spatial transformer networks (STN) [13, 21, 18] also aim at object-level transformations. Different from them which assume rigid transformations, with our method different pixel-level transformations are possible as depicted in the training data. This is desirable to handle articulated/deformable objects and self-occlusions. Moreover, our method does not depend on expensive pixel-level supervision as in [21].

3. METHODOLOGY

In this section, we describe our model using the clothing try-on / take-off as an example. Our goal is to transfer the shape information while keeping the appearance information, all trained without access to paired data. For this, we propose the asymmetric two-stream model shown in Fig. 2. The asymmetry reflects the fact that one of the two domains (domain B) is object-focused (e.g. catalog images of clothing items) while the other one (domain A) shows the objects in context (e.g. pictures of clothed persons).

Here, we use x_A and x_B to refer to images from domain A and domain B respectively. x_{AB} refers to images transferred from domain A to domain B, and vice versa for x_{BA}.

3.1. Assumptions

In previous works [8, 34], the try-on and take-off tasks are solved in a supervised way, respectively. Here, we solve both tasks in one model using unpaired data based on shared-latent space and context-structure constraints. **Shared-latent space constraint.** Similar to [11, 19, 24], we decompose the latent space into a content space and a style space. Differently, we assume that both content and style latent spaces can be shared by the two domains. We use $\mathrm{Z}^{\mathrm{C}}_{\mathrm{A}}$ and $\mathrm{Z}^{\mathrm{C}}_{\mathrm{B}}$ to denote the content space of domains A and B, and use $\mathrm{Z}^{\mathrm{S}}_{\mathrm{A}}$ and $\mathrm{Z}^{\mathrm{S}}_{\mathrm{B}}$ to denote the style space of domain A and B, respectively. Note that the style information is already shared between domain A and B images, but the content information is not. Therefore, we use one weight-sharing encoder $E^{\mathrm{S}}_{\mathrm{shared}}$ to obtain the shared style space constraint, and two encoders, $E^{\mathrm{C}}_{\mathrm{A}}$ and $E^{\mathrm{C}}_{\mathrm{B}}$, to achieve the shared content space constraint. **Context constraint.** The above shared-latent space constraints enable unpaired I2I and work well for style transfer tasks [23, 19, 24]. Yet it is not enough for geometry transfer when the output is multi-modal (i.e. multiple possible outputs). Here, we propose to use contextual guidance to constrain the output to be deterministic, i.e. decompose the one-to-many mapping into one-to-one mappings. In particular, for the try-on stream, we propose a Fit-in module which combines the feature maps with the context information. As to the take-off stream, we assume the output is unimodal and directly use the adversarial learning to learn the deterministic many-to-one mapping.

3.2. Network architecture

The model can be divided into try-on and take-off streams.

Try-on stream The catalog image x_{B} first passes through the domain B content encoder $E^{\mathrm{C}}_{\mathrm{B}}$ producing the content code $z^{\mathrm{C}}_{\mathrm{B}}$ in the shared content space $\mathrm{Z}^{\mathrm{C}}_{shared}$. In parallel, x_{B} is also encoded into a style code $z^{\mathrm{S}}_{\mathrm{B}}$ in the shared style space $\mathrm{Z}^{\mathrm{S}}_{\mathrm{shared}}$ by the shared style encoder E^{S}_{shared}. To combine the content and style information in the decoder, we use adaptive instance normalization (AdaIN) [10] layers for all residual and up-sampling blocks. The AdaIN parameters p_{AdaIN} are dynamically computed by a multi-layer perceptron from the style code $z^{\mathrm{S}}_{\mathrm{B}}$ to ensure the generated person image x_{BA} has the same style as x_{B}.

$$\mathrm{AdaIN}(z, \gamma, \beta) = \gamma \frac{(z - \mu(z))}{\delta(z)} + \beta \tag{1}$$

where z is the activation of the previous convolution layer. μ and δ are the mean and standard deviation computed per channel. Parameters γ and β are the output of the MLP of the shared style encoding module.

During decoding, the content code $z^{\mathrm{C}}_{\mathrm{B}}$ concatenated with the shape mask m_{A} are fed to the decoder G_{A}. There the content and style are fused by AdaIN and then fed to the Fit-in module. The Fit-in module is designed to enforce the context information constraint. We first estimate the bounding box of the mask from the context image. Then, we resize and align the up-sampled feature maps to this bounding box. Finally, this output is concatenated with the context image. The main goal of this design is to integrate the context information which helps the deterministic shape transform. The final try-on image x_{BA} is generated after the last convolution block.

In addition, we introduce an attention mechanism to both generator and discriminator. We concatenate the mask m_{A}

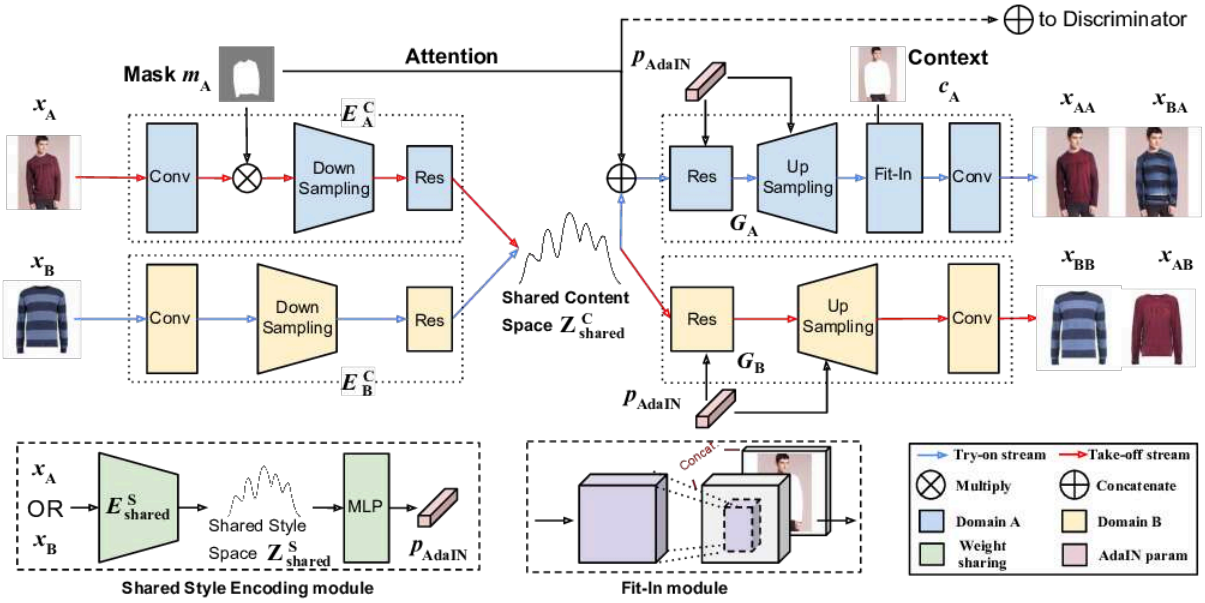

Fig. 2. Proposed unpaired Shape Transformer (UST). The try-on and take-off streams are trained jointly with shared style/content space constraints. To learn the one-to-many mapping in the try-on stream, the context information is utilized in the Fit-in module to constrain the output to be deterministic. Besides, an attention mechanism is applied to encourage the network to focus on the object. To learn the many-to-one mapping in the take-off stream, adversarial learning is adopted directly.

with the content code $z^{\mathrm{C}}_{\mathrm{B}}$ before the generator G_A and concatenate the mask m_{A} with the generated image x_{BA} before the discriminator D_A, respectively. This simple but effective attention operation encourages the network to focus on the generated clothing instead of the context part. This improves the results, especially when the objects to be translated have a highly variable scale/location within the images.

Take-off stream For the take-off stream, the clothed person image x_{A} first passes through a convolution block and then gets multiplied with the clothing mask m_{A} in order to exclude the background and skin information. Similar to the try-on stream, the masked feature maps are then encoded into a content code $z^{\mathrm{C}}_{\mathrm{A}}$ in the shared content space $\mathrm{Z}^{\mathrm{C}}_{\mathrm{shared}}$.

For the decoding part, the only difference with the try-on stream is that there is no "Fit-in" module or mask attention.

3.3. Learning

In this section, we only describe A→B translation for simplicity and clarity. The B→A is learned in a similar fashion. We denote the content latent code as $\mathrm{z}^{\mathrm{C}}_{\mathrm{A}}{=}E^{\mathrm{C}}_{\mathrm{A}}(x_{\mathrm{A}})$, style latent code as $\mathrm{z}^{\mathrm{S}}_{\mathrm{A}}{=}E^{\mathrm{S}}_{\mathrm{shared}}(x_{\mathrm{A}})$, within domain reconstruction output as $x_{\mathrm{AA}}{=}G_{\mathrm{A}}(\mathrm{z}^{\mathrm{C}}_{\mathrm{A}}, \mathrm{z}^{\mathrm{S}}_{\mathrm{A}})$, cross domain translation output as $x_{\mathrm{AB}}{=}G_{\mathrm{B}}(\mathrm{z}^{\mathrm{C}}_{\mathrm{A}}, \mathrm{z}^{\mathrm{S}}_{\mathrm{A}})$. Our loss function contains terms for the bidirectional reconstruction loss, cycle-consistency loss and adversarial loss [11, 19]. Besides, we also use a composed perceptual loss to preserve the appearance information across domains, and a symmetry loss capturing some extra domain knowledge [9, 36].

Bidirectional reconstruction loss ($L^{x_A}_{LR}$, $L^{x_A}_{SR}$). This loss consists of the feature level latent reconstruction loss $\mathcal{L}_{\mathrm{LR}}$ and the pixel level image self-reconstruction loss $\mathcal{L}_{\mathrm{SR}}$. The former contains both content and style code reconstructions.

$$\begin{aligned}\mathcal{L}^{x_{\mathrm{A}}}_{\mathrm{LR}} =& \mathbb{E}_{x_{\mathrm{AB}}, z^{\mathrm{C}}_{\mathrm{A}}}\left[\|E^{\mathrm{C}}_{\mathrm{B}}(x_{\mathrm{AB}}) - \mathrm{z}^{\mathrm{C}}_{\mathrm{A}}\|_1\right] \\ &+\mathbb{E}_{x_{\mathrm{AB}}, z^{\mathrm{S}}_{\mathrm{A}}}\left[\|E^{\mathrm{S}}_{\mathrm{shared}}(x_{\mathrm{AB}}) - \mathrm{z}^{\mathrm{S}}_{\mathrm{A}}\|_1\right] \quad (2)\\ \mathcal{L}^{x_{\mathrm{A}}}_{\mathrm{SR}} =& \mathbb{E}_{x_{\mathrm{A}}}\left[\|x_{\mathrm{AA}} - x_{\mathrm{A}}\|_1\right], \quad (3)\end{aligned}$$

Adversarial loss ($L^{x_A}_{GAN}$). To make the translated image look domain realistic, we use an adversarial loss to match the domain distribution.

$$\mathcal{L}^{x_{\mathrm{A}}}_{GAN} = \mathbb{E}_{x_{\mathrm{B}}}\left[\log D_{\mathrm{B}}(x_{\mathrm{B}})\right] + \mathbb{E}_{x_{\mathrm{AB}}}\left[\log\left(1 - D_{\mathrm{B}}(x_{\mathrm{AB}})\right)\right] \quad (4)$$

Cycle-consistency loss ($L^{x_A}_{CC}$). To enable unpaired translation, the cycle-consistency loss [38] is applied to stabilize the adversarial training.

$$\mathcal{L}^{x_{\mathrm{A}}}_{\mathrm{CC}} = \mathbb{E}_{x_{\mathrm{AB}}, x_{\mathrm{A}}}\left[\|G_{\mathrm{A}}(E^{\mathrm{C}}_{\mathrm{B}}(x_{\mathrm{AB}}), E^{\mathrm{S}}_{\mathrm{shared}}(x_{\mathrm{AB}})) - x_{\mathrm{A}}\|_1\right] \quad (5)$$

Perceptual loss ($L^{x_A}_{P}$) To preserve the appearance information, we apply a composed perceptual loss.

$$\begin{aligned}\mathcal{L}^{x_{\mathrm{A}}}_{\mathrm{P}} =& (\mathbb{E}_{x_{\mathrm{AA}}, x_{\mathrm{A}}}\left[\|\Phi(x_{\mathrm{AA}}) - \Phi(x_{\mathrm{A}})\|^2_2\right]) \\ &+(\mathbb{E}_{x_{\mathrm{AB}}, x_{\mathrm{B}}}\left[\|\Phi(x_{\mathrm{AB}}) - \Phi(x_{\mathrm{A'}})\|^2_2\right]) \\ &+\lambda\, \mathbb{E}_{x_{\mathrm{AB}}, x_{\mathrm{B}}}\left[\|Gram(x_{\mathrm{AB}}) - Gram(x_{\mathrm{A'}})\|_1\right], \quad (6)\end{aligned}$$

where $x_{A'}$ is the Region of Interest (RoI) of x_A. For clothing items, it is the segmented clothing region. Φ is a network trained on external data, whose representation can capture image similarity. Similar to [5, 15], we use the first convolution layer of all five blocks in VGG16 [30] to extract the feature maps to calculate the Gram matrix that contains non-localized style information. λ is the corresponding loss weight.

Symmetry loss ($L_{Sym}^{x_A}$). To utilize the inherent prior knowledge of clothing, we apply a symmetry loss [9, 36] to the take-off stream.

$$\mathcal{L}_{\text{Sym}}^{x_{\text{A}}} = \mathbb{E}_{x_{\text{AB}}}\big[\frac{1}{W/2 \times H}\sum_{w=1}^{W/2}\sum_{h=1}^{H}\|x_{\text{AB}}^{w,h} - x_{\text{AB}}^{W-(w-1),h}\|_1\big], \quad (7)$$

where H and W denote the height and width of the image, (w, h) are the coordinates of each pixel, and $x_{\text{AB}}^{w,h}$ refers to a pixel in the transferred image x_{AB}.

Total loss. Our model, including encoders, decoders and discriminators, is optimized jointly. The full objective is as follows,

$$\begin{aligned}&\min_{\substack{E_{\text{A}}^{\text{C}}, E_{\text{B}}^{\text{C}}, E_{\text{shared}}^{\text{S}},\\ G_{\text{A}}, G_{\text{B}}}} \max_{D_{\text{A}}, D_{\text{B}}} \mathcal{L}(E_{\text{A}}^{\text{C}}, E_{\text{B}}^{\text{C}}, E_{\text{shared}}^{\text{S}}, G_{\text{A}}, G_{\text{B}}, D_{\text{A}}, D_{\text{B}})\\ &= \mathcal{L}_{\text{GAN}}^{x_{\text{A}}} + \mathcal{L}_{\text{GAN}}^{x_{\text{B}}} + \lambda_{\text{CC}}(\mathcal{L}_{\text{CC}}^{x_{\text{A}}} + \mathcal{L}_{\text{CC}}^{x_{\text{B}}}) + \lambda_{\text{SR}}(\mathcal{L}_{\text{SR}}^{x_{\text{A}}} + \mathcal{L}_{\text{SR}}^{x_{\text{B}}})\\ &+ \lambda_{\text{LR}}(\mathcal{L}_{\text{LR}}^{x_{\text{A}}} + \mathcal{L}_{\text{LR}}^{x_{\text{B}}}) + \lambda_{\text{P}}(\mathcal{L}_{\text{P}}^{x_{\text{A}}} + \mathcal{L}_{\text{P}}^{x_{\text{B}}}) + \lambda_{\text{Sym}}\mathcal{L}_{\text{Sym}}^{x_{\text{A}}}. \quad (8)\end{aligned}$$

4. EVALUATION

Datasets We evaluate our method on the clothing try-on and take-off tasks on the FashionStyle and VITON [8] dataset. VITON has around 16,000 images for each domain. However, we find that there are plenty of image duplicates with different file names. After cleaning the dataset, there are 7,240 images in each domain left. The FashionStyle dataset, provided by an industrial partner, has 5,230 training images and 1,320 testing images of clothed people (domain A), and 2,837 training images and 434 testing images of the clothing catalog items (domain B). For domain A, FashionStyle has multiple views of the same person wearing the same clothing item. We present results on this dataset for one category, namely pullover/sweater.

Metrics We use paired images from different domains depicting the same clothing item to quantitatively evaluate the performance of our method. For the case of the try-on task we measure the similarity between the RoI of the original image (from domain A) and the RoI of a generated version (where its corresponding clothing item has been translated to fit in a masked out version of the image). Thus, we call it *Try-on RoI*. To create this masked image we first perform clothing-item segmentation [20] to remove the clothing-item originally worn by the person. For the case of the take-off task, given an image from domain A, we measure the similarity of its corresponding clothing item (from domain B) with the generated item. On both cases similarity between images is computed using the SSIM [33] and LPIPS [35] metrics. We report the mean similarity across the whole testing set.

Implementation details The LPIPS [35] network is used as the perceptual feature extractor Φ in Eq. 6. In all our experiments, we use the Adam [16] optimizer with β_1=0.5 and β_2=0.999. The initial learning rate is set to 2×10^{-6}. Models are trained with a minibatch of size 1. We use the segmentation method [20] to get the clothing mask and its bounding box. The shared content code is a tensor whose dimension is determined by the data. The shared style code is a vector, we use 8 dimensions in our experiments.

Table 1. Mean SSIM and LPIPS-VGG similarity. Higher SSIM values and lower LPIPS indicate higher similarity.

Method	Try-on RoI (SSIM/LPIPS)	Take off (SSIM/LPIPS)
Ours	66.42 / **27.02**	**61.19 / 34.37**
Supervised model	***69.51 / 24.14***	***61.54 / 32.56***

Table 2. Comparisons w.r.t. state-of-the-art methods for the take-off task on the FashionStyle dataset.

Method	Original (SSIM/LPIPS)	with mask (SSIM/LPIPS)
CycleGAN	45.63 / 47.47	47.18 / 49.94
MUNIT	45.97 / 46.53	51.92 / 48.16
Ours	N/A	***61.19 / 34.37***

Fig. 3. Try-on and take-off results on the VITON dataset. For try-on (top) each column shows a person (from the top row) virtually trying on different clothing items. For take-off (bottom) each example consists of three images: input image, generated take-off image and the ground-truth (GT) image. Zoom in for more details.

4.1. Clothing try-on / take-off on FashionStyle

We present quantitative results on the translation performance of the try-on / take-off tasks in Table 1 for the FashionStyle dataset with related qualitative results presented in Fig. 1. The results indicate our method can preserve the color and patterns well in an unpaired way for both try-on and take-off tasks. Our Fit-in module does help the model tackle the one-to-many mapping problem. Please refer to supplementary material for more results.

4.2. Clothing try-on / take-off on VITON

We complement the previous results with a qualitative experiment (see Fig. 3) on the VITON dataset using the full model. We see that our method is able to effectively translate the shape of the clothing items across the domains. It is notable that on the try-on task, it is able to preserve the texture information of the items even in the presence of occlusions caused by arms. This is handled by the proposed Fit-in module (Sec. 3) which learns how to combine foreground and contextual information.

Fig. 4. Comparisons with CycleGAN [38] and MUNIT [11] for try-on (left) and take-off (right) on FashionStyle dataset.

Fig. 5. Comparison with VITON [8] and CP-VITON [31] (both supervised) on the *Try-on* task.

4.3. Comparisons with existing methods

We compare our model w.r.t. CycleGAN [38], MUNIT [11], VITON [8] and CP-VITON [31]. Fig. 4 shows qualitative results from our model, CycleGAN and MUNIT. It is clear that these unpaired methods cannot handle the one-to-many shape transfer task. CycleGAN can only work for one-to-one mapping task. MUNIT has the ability to do many-to-many mapping for style translation but it is unable to transfer shapes. We present quantitative results in Table 2. We report performance using the *original* version of those methods and a variant where the same *mask* used in our method is applied to their input. We do not provide the Try-on RoI scores since these existing methods cannot determine the RoI for Try-on images (see Fig. 4). The comparison with the supervised VITON methods is shown in Fig. 5. It is motivating that even without any supervised paired data, our method achieves competitive results.

5. CONCLUSION

We present a method to translate the shape of an object across different domains while preserving its style/appearance. Our experiments show that our method is able to achieve the task at hand while surpassing the performance of existing efforts.

Acknowledgements: This work was funded by the Agentschap Innoveren en Ondernemen (VLAIO) project HBC.2017.0358.

6. REFERENCES

[1] A. Almahairi, S. Rajeswar, A. Sordoni, P. Bachman, and A. Courville. Augmented cyclegan: Learning many-to-many mappings from unpaired data. In *ICML*, 2018.

[2] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein generative adversarial networks. In *ICML*, 2017.

[3] Y. Cao, Z. Zhou, W. Zhang, and Y. Yu. Unsupervised diverse colorization via generative adversarial networks. In *ECML/PKDD*, 2017.

[4] M. Frid-Adar, E. Klang, M. Amitai, J. Goldberger, and H. Greenspan. Synthetic data augmentation using gan for improved liver lesion classification. In *ISBI*, 2018.

[5] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *CVPR*, 2016.

[6] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio. Generative adversarial nets. In *NIPS*, 2014.

[7] L. Guo, J. Liu, Y. Wang, Z. Luo, W. Wen, and H. Lu. Sketch-based image retrieval using generative adversarial networks. In *ACM Multimedia*, 2017.

[8] X. Han, Z. Wu, Z. Wu, R. Yu, and L. S. Davis. Viton: An image-based virtual try-on network. In *CVPR*, 2018.

[9] R. Huang, S. Zhang, T. Li, R. He, et al. Beyond face rotation: Global and local perception gan for photorealistic and identity preserving frontal view synthesis. In *ICCV*, 2017.

[10] X. Huang and S. J. Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 2017.

[11] X. Huang, M. Liu, S. Belongie, and J. Kautz. Multimodal unsupervised image-to-image translation. In *ECCV*, 2018.

[12] P. Isola, J. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017.

[13] M. Jaderberg, K. Simonyan, A. Zisserman, and k. kavukcuoglu. Spatial transformer networks. In *NIPS*. 2015.

[14] Y. Jing, Y. Yang, Z. Feng, J. Ye, and M. Song. Neural style transfer: A review. *CoRR*, abs/1705.04058, 2017.

[15] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016.

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

[17] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. P. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi. Photo-realistic single image super-resolution using a generative adversarial. In *CVPR*, 2015.

[18] D. Lee, S. Liu, J. Gu, M.-Y. Liu, M.-H. Yang, and J. Kautz. Context-aware synthesis and placement of object instances. In *NIPS*, 2018.

[19] H.-Y. Lee, H.-Y. Tseng, J.-B. Huang, M. Singh, and M.-H. Yang. Diverse image-to-image translation via disentangled representations. In *ECCV*, 2018.

[20] X. Liang, K. Gong, X. Shen, and L. Lin. Look into person: Joint body parsing & pose estimation network and a new benchmark. *TPAMI*, 2018.

[21] C.-H. Lin, E. Yumer, O. Wang, E. Shechtman, and S. Lucey. St-gan: Spatial transformer generative adversarial networks for image compositing. In *CVPR*, 2018.

[22] G. Liu, J. Wang, C. Zhang, S. Liao, and Y. Liu. Realistic view synthesis of a structured traffic environment via adversarial training. In *CAC*, 2017.

[23] M.-Y. Liu, T. Breuel, and J. Kautz. Unsupervised image-to-image translation networks. In *NIPS*, 2017.

[24] L. Ma, X. Jia, S. Georgoulis, T. Tuytelaars, and L. Van Gool. Exemplar guided unsupervised image-to-image translation with semantic consistency. *ICLR*, 2019.

[25] L. Ma, X. Jia, Q. Sun, B. Schiele, T. Tuytelaars, and L. Van Gool. Pose guided person image generation. In *NIPS*, 2017.

[26] L. Ma, Q. Sun, S. Georgoulis, L. Van Gool, B. Schiele, and M. Fritz. Disentangled person image generation. In *CVPR*, 2018.

[27] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. P. Smolley. Least squares generative adversarial networks. In *ICCV*, 2017.

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

[29] A. Raj, P. Sangkloy, H. Chang, J. Lu, D. Ceylan, and J. Hays. Swapnet: Garment transfer in single view images. In *ECCV*, 2018.

[30] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[31] B. Wang, H. Zhang, X. Liang, Y. Chen, L. Lin, and M. Yang. Toward characteristic-preserving image-based virtual try-on network. In *ECCV*, 2018.

[32] B. Wang, H. Zheng, X. Liang, Y. Chen, L. Lin, and M. Yang. Toward characteristic-preserving image-based virtual try-on network. In *ECCV*, 2018.

[33] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *Trans. Img. Proc.*, 2004.

[34] D. Yoo, N. Kim, S. Park, A. S. Paek, and I. Kweon. Pixel-level domain transfer. In *ECCV*, 2016.

[35] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

[36] J. Zhao, Y. Cheng, Y. Xu, L. Xiong, J. Li, F. Zhao, K. Jayashree, S. Pranata, S. Shen, J. Xing, S. Yan, and J. Feng. Towards pose invariant face recognition in the wild. In *CVPR*, 2018.

[37] C. Zheng, T. Cham, and J. Cai. T ˆ2 2 net: Synthetic-to-realistic translation for solving single-image depth estimation tasks. In *ECCV*, 2018.

[38] J. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017.