Article

Would Your Clothes Look Good on Me? Towards Transferring Clothing Styles with Adaptive Instance Normalization

Tomaso Fontanini * and Claudio Ferrari

Department of Engineering and Architecture, University of Parma, 43124 Parma, Italy; claudio.ferrari2@unipr.it
* Correspondence: tomaso.fontanini@unipr.it

Abstract: Several applications of deep learning, such as image classification and retrieval, recommendation systems, and especially image synthesis, are of great interest to the fashion industry. Recently, image generation of clothes gained lots of popularity as it is a very challenging task that is far from being solved. Additionally, it would open lots of possibilities for designers and stylists enhancing their creativity. For this reason, in this paper we propose to tackle the problem of style transfer between two different people wearing different clothes. We draw inspiration from the recent StarGANv2 architecture that reached impressive results in transferring a target domain to a source image and we adapted it to work with fashion images and to transfer clothes styles. In more detail, we modified the architecture to work without the need of a clear separation between multiple domains, added a perceptual loss between the target and the source clothes, and edited the style encoder to better represent the style information of target clothes. We performed both qualitative and quantitative experiments with the recent DeepFashion2 dataset and proved the efficacy and novelty of our method.

Keywords: deep learning; style transfer

1. Introduction

Recently, deep learning has drawn lots of attention from the fashion industry due to its possible application to many different tasks. These include the automatic classification of clothes [1], which can be employed, for instance, for the automatic sorting of an online shop. Next, retrieval techniques can be used in fashion [2] to help the client find a specific item given a single picture. In addition, a good recommendation system is fundamental for basically every online shop, especially fashion ones [3], where keeping track of the user’s preferences and presenting suggestions to the client based on these can help improve sales by a great margin.

Even if all these tasks present several unresolved challenges, one that is still in an early stage regards synthesizing clothes images using deep learning. Particularly, this is due to the impressive amount of variability that characterizes the clothes image category. Clothes exist in significantly large variety of styles, shapes, textures, materials, and colors; this represents a great obstacle, even for state-of-the-art generative techniques, as in [4]. Indeed, it can be claimed that is much easier for a network to generate a realistic face than realistic clothes. Nevertheless, being able to generate realistic clothes can be of great interest for designers and stylists because it can greatly help their work and boost their creativity.

Regardless of the great challenge that this particular task brings with it, several works were presented to tackle this problem from different angles. Some works had the objective of generating a full person wearing particular clothes, as in [5]. This process was sometimes facilitated by starting the generation from a segmentation map [6]. In this case, the task is more similar to the standard semantic image synthesis with generative adversarial networks (GANs). However, it could have less applicability in the fashion industry due to the lack of control on what the network will generate.
In addition to that, recently, virtual try-on systems have drawn lot of attention [7,8]. They have the objective of fitting particular clothes on a target image of a person. They can be very useful for simulating wearing particular clothes before buying and, for this reason, are especially relevant for online shops where users do not have the chance to try a particular clothes. More specifically, virtual try-on systems usually rely on fitting specific source clothes to a target mask and completely replacing them.

In this work, we address the following question: given two images of people, one target and one source, wearing some clothes, is it possible to transfer the style (color, texture) of the target’s clothes to the source image while maintaining the original shape, e.g., dress, pants, shirt? Differently from existing works (such as [9]) which extract styles from random images such as paintings or textures and try to apply it to a single clothes image (with no corresponding person wearing it), we opt for a more challenging setting in which we extract the “clothing style” from a target person and transfer it to a source person. This is of great interest for both customer and industry, as the first has the possibility to explore different styles of clothes matching his or her taste and the latter can generate different clothes prototypes without the need of making them manually. To deal with the fact both source and target images contain people, we use a segmentation mask to separate the clothes area with the rest of the image that does not have to be considered during the style transfer.

This is performed by employing a style encoder that extracts style codes from the clothes area, and then injects these codes into a generator network consisting of an encoder–decoder structure. In more detail, the style codes are used to produce parameters for the Adaptive Instance Normalization (AdaIN) [10] layers that are present in the decoder network. Indeed, AdaIN layers allows us to transfer the target style to the source image without additional training steps during inference time. Our network architecture takes inspiration from StarGANv2 [11], though with some fundamental changes; in particular, we remove the domain separations in the model as it is not important for our style transfer task. Then, we modified the StarGANv2 style encoder architecture augmenting the style dimension and replacing two downsampling layers with a pooling function in order to remove the shape information from the style codes.

To summarize, the main contributions of the proposed work are the following:

- A system that is able to extract a “clothing style” from a person wearing a set of clothes and transfer it to a source person without additional training steps required during inference;
- The model is trained in an end-to-end matter and employs clothes masks in order to localize the style transfer only to specific areas of the images, leaving the background and person’s identity untouched;
- Extensive experiments performed on a particularly-challenging dataset, i.e., Deep-Fashion2 [12].

The paper will be organized as follows: firstly, in Section 2, current state-of-the-art literature will be presented; then, in Section 3, the proposed architecture will be described, together with the training losses; Section 4 reports experimental results, both qualitatively and quantitatively; finally, in Section 5 conclusions and future works will be discussed.

2. Related Work

Generative Adversarial Networks. In recent years, Generative Adversarial Networks (GANS) have become the state-of-the-art for generative tasks. They were first introduced by Ian Goodfellow et al. in [13] and were quickly adopted for multiple tasks. For instance, in noise-to-image generation, the objective is to sample images starting from a random distribution $z$. In this field, DCGAN [14] presented the first fully convolutional GAN architecture. In addition, BiGAN [15] introduced an inverse mapping for the generated image, and conditional GANs [16] were introduced in order to control the output of a GAN using additional information. Then, BigGAN [17] proved that GANS could successfully generate high-resolution, diverse samples. Recently, StyleGAN [18] and StyleGANv2 [4]...
generated style codes starting from the noise distribution and used them to guide the generation process, reaching impressive results. Indeed, GANs can be employed to solve several challenges such as text-to-image generation [19], sign image generation [20] or removing masks from faces [21]. Another task in which GANs can be used with great success is image-to-image translation, where an image is mapped from a source domain to a target domain. This can be performed both in a paired [22] or an unpaired [23] way. In addition, StarGAN [24] proposed a solution in order to use a single generator even when transferring multiple attributes. However, of these works can be adapted to perform arbitrary style transfer in fashion, as they do not employ style codes as part of the generation process and focus more on performing manipulation over a small set of domains. Finally, with the introduction of Adaptive Instance Normalization [10], several GANs architectures were proposed to perform domain transfer. Some examples are MUNIT [25], FUNIT [26], or StarGANv2 [11]. In particular, StarGANv2 is able to generate images starting from a style extracted from an image or generated with a latent code and represents the main baseline for this work. Nevertheless, these methods are not specifically designed for fashion style transfer, as they also apply changes to the shape of the input image. For this reason, in this paper, we chose to adapt StarGANv2 architecture to perform this new task.

Neural Style Transfer. Style transfer has the objective of transferring the style of a target image on a source image, leaving its content untouched. Gatys et al. [27] were the first to use a convolutional neural network to tackle this task. Then, Refs. [28,29] managed to solve the optimization problem proposed by Gatys et al. in real-time. Since then, several other works were proposed [30–35]. Then, Chen and Schmid [36] were able to perform arbitrary style transfer, but their proposed method was very slow. None of these methods were tested with fashion images, and they are conceived for transferring styles typically from a painting to a picture and, therefore, cannot be used in this work. On the other hand, for this paper, the fundamental step is represented by the introduction of Adaptive Instance Normalization (AdaIN) [10] that, for the first time, allowed us to perform fast arbitrary style transfer in real-time without being limited to a specific set of styles as in previous works.

Deep Learning for Fashion. In the fashion industry, deep learning can be used in several applications. Firstly, the task of the classification of the clothing fashion styles was explored in several works [1,37,38]. Another fundamental topic is clothing retrieval [2,39–41], which can be used to find a specific clothes in an online shop using a picture taken in the wild. In addition to that, recommendation systems are also of great interest in order to suggest particular clothes based on the user’s preferences [3,42–47]. The task that is most relevant to this paper is that of synthesizing clothes images. This can be achieved by selecting a clothes image and generating an image of a person wearing the same clothes [5,48]. In addition, some works generated clothes starting from segmentation maps [49] or text [6]. Finally, virtual try-on systems have the objective of altering the clothes worn by a single person. Firstly, CAGAN [50] performed automatic swapping of clothing on fashion model photos using cycle-consistency loss. Next, VITON [7] generated a coarse-synthesized image with the target clothing item overlaid on the same person in the same pose. Then, the initial blurry clothing area is enhanced with a refinement network. In addition, Kim et al. [8] performed the try-on disentangling geometry and style of the target and source image, respectively. Furthermore, Jiang et al. ([9,51]) used a spatial mask to restrict different styles to different areas of a garment. Finally, Lewis et al. [52] employed a pose conditioned StyleGAN2 architecture and a layered latent space interpolation method.

Differently from previous methods, our model employs a customized version of StarGANv2 [11]. In addition, the style is not extracted from an art image or a texture, but it is taken directly from target clothes, making the process much more challenging.

3. Proposed System

In order to design a model that is able to transfer a clothing style from a source image to a target image, we took inspiration from StarGANv2 [11], but made some modification
to its architecture and training. We will describe the architecture and the training in detail in the next sections.

3.1. Network Architecture

The proposed model is composed by a Generator $G$, a style encoder $E$, a mapping network $M$, and a Discriminator $D$ (see Figure 1). When transferring a style, $E$ takes as input a target image $x_{trg}$ and generates a style $s_{trg}$, while $G$ takes a source image $x_{src}$ and the style $s_{trg}$ as input generating an output image $\hat{x}_{src}$. The style can also be produced from a random noise $z$ that is used as input for the mapping network $M$ which produces a random style code that can be used by $G$ to generate a new image with a custom style. In addition, $D$ has the role of evaluating if the generated samples look real or fake, following the well-known GAN paradigm.

Figure 1. Overview of the proposed system. Firstly, the input image $x_{src}$ is multiplied with its corresponding mask $m_{src}$ and fed to the generator $G$. Then, AdaIN parameters are extracted from either $E$ or $M$ and injected in the normalization layers of $G$. Finally, the generator output is multiplied again with the inverted mask and used as input to the discriminator $D$.

More in detail, $G$, as in [11], is an encoder–decoder architecture with four downsampling residual blocks and four upsampling residual blocks. Additionally, the discriminator $D$ structure takes inspiration from the StarGANv2 discriminator. However, as several labeled ground-truth domains are not available, we could not take advantage from splitting the discriminator output to classify each domain independently and opted for designing $D$ to have a single output. Finally, the style encoder $E$ is a single convolutional encoder, while $M$ is a multilayer perceptron.

3.1.1. Style Encoder Architecture

Differently from the style encoder of StarGANv2, we heavily modified the architecture of $E$ (see Figure 2). In particular, the StarGANv2 encoder did not completely erase the shape information of the reference image, as it needs to change more than just the texture in the source image (such as hair style). On the contrary, we removed this information by inserting a pooling layer and removing two down-sampling layers in the network, as we only need to transfer the clothes style and its shape information is useless.
Figure 2. Proposed style encoder architecture. As opposed to StarGANv2, we removed shape information using a pooling function and removing the last two downsampling layers.

3.1.2. Transferring a Style with Adaptive Instance Normalization

In order to apply a target style to an image, $G$ makes use of AdaIN layers [10]. Specifically, as opposed to standard Instance Normalization (IN) layers, the mean $\mu$ and variance $\sigma$ of the input $x_{src}$ are aligned with ones of the style $s_{trg}$:

$$
    AdaIN(x_{src}, s_{trg}) = \sigma(s_{trg}) \left( \frac{x_{src} - \mu(x_{src})}{\sigma(x_{src})} \right) + \mu(s_{trg})
$$

where both $\sigma(s_{trg})$ and $\mu(s_{trg})$ are generated using the style encoder $E$ or the mapping network $M$.

Differently from StarGANv2, we only needed to apply the style on a localized area of the source image, which corresponds to the location of the clothes. For this reason, we employed a masking technique in order to achieve this goal. In particular, given masks $m_{trg}$ and $m_{src}$ correspond to the clothes area in the target and source image, respectively, the style transfer equation for our system becomes:

$$
    \hat{x}_{src} = G(m_{src} \cdot x_{src}, E(m_{trg} \cdot x_{trg})) + (1 - m_{src}) \cdot x_{src}
$$

In more detail, both source and target images are masked in order to apply the style and extract it from only the clothes area, and after the translation the background is applied again to the output images. Indeed, clothes masks are provided with the DeepFashion2 [12] dataset, but they can be easily extracted from any image using a human parsing network, as in [53]. Finally, masking the images also has the effect of helping the generalization capabilities of the model, as the network does not have to learn all the possible backgrounds and can tune its parameters only for learning the multiple possible clothes shapes and styles.

3.2. Training

In order to train the proposed model, we borrow the training scheme from StarGANv2 and improve it in order to enhance its style transfer capability. First of all, an adversarial loss $L_{adv}$ is employed in order to generate realistic results. Differently from StarGANv2, our discriminator only needs to decide if clothes look real or fake, regardless of its category:

$$
    L_{adv} = E_{x_{src}}[\log D(x_{src})] + E_{x_{trg}}[\log (1 - D(x_{src}))]
$$
Secondly, a style reconstruction loss $\mathcal{L}_{\text{sty}}$ is introduced in order to prevent the generator $G$ from ignoring the style code $s_{\text{trg}}$. It is defined as the distance between the style code $s_{\text{trg}}$ and the style code extracted from the generated image $\hat{x}_{\text{src}}$:

$$
\mathcal{L}_{\text{sty}} = \mathbb{E}_{x_{\text{src}},x_{\text{trg}}} [\| s_{\text{trg}} - E(\hat{x}_{\text{src}}) \|]
$$

(4)

Thirdly, a style diversification loss $\mathcal{L}_{\text{div}}$ serves the purpose of enforcing diversity in the generated outputs (two different style codes should always produce different images):

$$
\mathcal{L}_{\text{div}} = \mathbb{E}_{x_{\text{src}},x_{\text{trg}},x_{\text{trg}}_1,x_{\text{trg}}_2} [\| \hat{x}_{\text{src}}_1 - \hat{x}_{\text{src}}_2 \|]
$$

(5)

In this case, the style codes are produced starting from two different random noises, $z_1$ and $z_2$, fed to the mapping network $M$. Finally, a cycle consistency loss $\mathcal{L}_{\text{cyc}}$ avoids the source image to change its domain-invariant characteristics such as shape and pose:

$$
\mathcal{L}_{\text{cyc}} = \mathbb{E}_{x_{\text{src}},x_{\text{trg}}} [\| x_{\text{src}} - [G(m_{\text{src}} \cdot \hat{x}_{\text{src}}, E(m_{\text{src}} \cdot x_{\text{src}})) + (1 - m_{\text{src}}) \cdot \hat{x}_{\text{src}}] \|]
$$

(6)

In addition to these loss functions, we introduced a new loss term to train the network. In particular, we took advantage of a perceptual loss $\mathcal{L}_{\text{prc}}$ \cite{55} between the target masked image $m_{\text{trg}} \cdot x_{\text{trg}}$ and the source masked generated samples $m_{\text{src}} \cdot \hat{x}_{\text{src}}$. The reason is to force the output clothes to be perceptually similar to the target clothes. As a consequence, we managed to perform a better style transfer. Perceptual loss can be written as follows:

$$
\mathcal{L}_{\text{prc}} = \psi(m_{\text{src}} \cdot \hat{x}_{\text{src}}, m_{\text{trg}} \cdot x_{\text{trg}})
$$

(7)

where $\psi$ is the perceptual distance which is calculated as the difference between a series of weighted activation channels of a pretrained network such as VGG19 \cite{55}. The advantage of a perceptual loss is that it is not linked to the pixel difference between the two images, which would be too strong a condition for our purposes. Finally, the full objective function becomes:

$$
\min_{G,E} \max_D \mathcal{L}_{\text{adv}} + \lambda_{\text{sty}} \mathcal{L}_{\text{sty}} - \lambda_{\text{div}} \mathcal{L}_{\text{div}} + \lambda_{\text{cyc}} \mathcal{L}_{\text{cyc}} + \lambda_{\text{prc}} \mathcal{L}_{\text{prc}}
$$

(8)

where $\lambda_{\text{sty}}$, $\lambda_{\text{div}}$, $\lambda_{\text{cyc}}$, and $\lambda_{\text{prc}}$ are the weights of the various losses.

4. Results and Discussion

In this section, all experiments performed with the proposed architecture will be presented and discussed. Both qualitative and quantitative results will be evaluated with comments.

4.1. Dataset

To train the system architecture, we choose to utilize the DeepFashion2 dataset \cite{12}. It is composed of 491 k images of people wearing 13 categories of clothes: short sleeve top, long sleeve top, short sleeve outwear, long sleeve outwear, vest, sling, shorts, trousers, skirt, short sleeve dress, long sleeve dress, vest dress, and sling dress. The dataset provides additional information for each clothes item such as scale, occlusion, zoom-in, viewpoint, category, style, bounding box, dense landmarks, and per-pixel mask. Images are also divided into commercial and customer types and are taken from several viewpoints and orientations, making it a particularly difficult dataset for a neural network to analyze.

In Figure 3, some images extracted from the dataset are shown. The segmentation mask is the most important feature for our system as it allows us to extract only the style from the target clothes area and apply it only to the source clothes area.
Figure 3. Some images from DeepFashion2 dataset. For each image, a segmentation mask of the clothes is given.

4.2. Network Configuration and Parameters

During all the experiments, we trained the model for 50k iterations on an RTX Nvidia Quadro 4000. Training took about 2 days. As an optimizer, we used Adam [56] with a learning rate of 0.0001. During the training, we fixed the parameters as expressed in Table 1. In particular, we chose to employ a style code dimension of 512 instead of 64 (default dimension for StarGANv2) because we empirically found that a higher style dimension is able to express and transfer the style more consistently.

Table 1. Parameters employed during training.

| Training Parameters | Value |
|---------------------|-------|
| Style dimension     | 512   |
| $\lambda_{sty}$     | 1     |
| $\lambda_{div}$     | 1     |
| $\lambda_{cyc}$     | 1     |
| $\lambda_{prc}$     | 10    |

4.3. Experiments

We first present some qualitative results in Figure 4. It can be seen how our method is able to apply the style of the target image over the source image even in the very challenging setting of DeepFashion2 dataset. Our system can especially transfer the color style, still being aware of some local information: for example, in the second triplet of the third row, the source image presents a black shirt and a white skirt but, as the target image girl wears a white shirt and black pants, the colors are inverted in the output image. Furthermore, in the first triplet of the third row, the style image contains a light colored shirt and a blue skirt which are then applied in this fashion over the source image. This would not be possible with other older classical computer vision methods such as, for example, histogram matching.

On the contrary, the proposed model still struggles when transferring texture. This is due to the global nature of the AdaIN layers. Indeed, these kinds of normalization layers are not able to apply precise changes over the images and, for this reason, small details in the target images are not transferred over the source images. This is particularly evident for clothes with text or small objects printed on them. Nevertheless, despite not being able to exactly replicate a texture, if the target clothes contain complex texture, the model is still able to generate a result that is not simply a plain color. Indeed, solving this problem can be an interesting future work.
Figure 4. Some results with the proposed method.

After proving the efficacy of our work, in Figure 5 we demonstrate that the baseline StarGANv2 is not suited for the task tackled in this paper. Indeed, StarGANv2 uses a different style encoder network and a smaller style dimension. Looking at the figure, it is clear how it is not able to correctly transfer the clothes style and leaves the output almost identical to the source image. The reason for this is that, given the impressive intrinsic variety of the DeepFashion2 dataset in terms of styles, pose, camera angles, and zoom, the model can not generalize enough in order to produce a meaningful style code that is basically ignored by the generator $G$. On the other side, with the customizations that we introduced, our model overcomes this issue.
Regarding quantitative evaluation, we present the results in Table 2. As in other works on style transfer, we adopted the Learned Perceptual Image Patch Similarity (LPIPS) distance between the source clothes and the generated samples. LPIPS is a metric to evaluate the perceptual similarity of images. It was firstly introduced in [54] for evaluating style transfer methods, and it is based on the idea of comparing intermediate features of deep networks. In [54], several drawbacks of standard measures such as Structural Similarity (SSIM) or Peak Signal to Noise Ratio (PSNR) were identified. It was shown how LPIPS is more reliable and more accurately resembles human perception of similarity between images while being robust to noise. On the contrary, we chose not to use the popular quantitative image quality evaluation metric FID score [57], as our method performs a heavy editing over the images (while StarGANv2 leaves them almost untouched) and this would not be grasped by FID.

Table 2. Quantitative comparison between StarGANv2 and our method. Results with and without background are reported separately. (Bold represent the best result.)

| Method        | LPIPS ↑ |
|---------------|---------|
| All Image     |         |
| StarGANv2 [11]| 0.524   |
| Ours          | 0.551   |
| Clothes Area  |         |
| StarGANv2 [11]| 0.162   |
| Ours          | 0.267   |

Given that our method transfers the style, i.e., manipulates only a region of the image (the clothes), we computed the LPIPS measure for both (i) the whole images and (ii) the clothes area only. This is intended to highlight the difference in the generated styles. The results in Table 2 indeed show that, with respect to StarGANv2, our method successfully
applies a style that makes the generated clothes different from the originals. In addition, in Figure 6 we present some results when the style code is generated starting from a latent code $z$ using the mapping network $M$. It is clear how our model is able to generate very different and diverse styles, both in terms of color and motives. Indeed, being able to generate style from random noise is a feature that allows lots of additional freedom in generation and can be of great interest for designers.

Figure 6. Different styles generated with the mapping network $M$. 
In Figure 7, we report some additional qualitative examples using a new dataset. Specifically, we kept the original network weights trained with DeepFashion2 and performed an inference step using images from the original DeepFashion dataset [41]. Indeed, as the DeepFashion dataset does not provide segmentation masks for the clothes area, we generated them using a pretrained human parsing network [33].

![Image](image.png)

Figure 7. Results with DeepFashion [41] dataset.

4.4. Ablation

Finally, we performed several ablations in order to find the best configuration for our method. Table 3 reports the most significant ones. Firstly, we show how with the StarGANv2 style encoder the perceptual distance between source clothes and output decreases. This is due to the better capability of our encoder of transferring the style to the source image. Secondly, we removed the cycle consistency loss to see if $L_{prc}$ would be enough to train the model. Experiments proved that the cycle consistency loss is very important for maintaining the high quality of the style transfer, as it stabilizes the training and maintains the source content and textures during the style transfer. Thirdly, we removed the perceptual loss and, as expected, by doing so the network started struggling when applying the target style, as the generated sample clothes are not bounded to be perceptually similar to the target clothes anymore. Finally, the weight of $L_{prc}$ was reduced, which resulted in worse performance overall.

|                      | LPIPS ↑  |
|----------------------|---------|
| Ours                 | 0.551   |
| StarGANv2 Encoder    | 0.540   |
| no $L_{cyc}$         | 0.542   |
| no $L_{prc}$         | 0.539   |
| $\lambda_{prc} = 1$  | 0.545   |

5. Conclusions

In this paper, we presented a novel way to transfer a style extracted from target clothes to a source image containing a person wearing different clothes. This was achieved taking inspiration from the StarGANv2 architecture, but customizing it in order to perform this task. In particular, the style dimension was bigger; the style encoder network was modified, removing two downsampling and adding a pooling layer; and the concept of multiple domains was removed.

Experiments were performed with the challenging DeepFashion2 dataset and proved the efficacy of our method. Nevertheless, results were still not ideal, as the network is not
able to correctly transfer texture information and small objects printed on the target clothes due to how AdaIN layers work.

Future works include solving these issues and extending the network to a wider variety of datasets and tasks.

**Author Contributions:** Conceptualization, T.F. and C.F.; methodology, T.F.; software, T.F.; validation, T.F.; formal analysis, T.F.; investigation, T.F.; resources, T.F.; data curation, T.F.; writing—original draft preparation, T.F.; writing—review and editing, C.F.; visualization, T.F.; supervision, C.F.; project administration, C.F. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not Applicable.

**Informed Consent Statement:** Not Applicable.

**Data Availability Statement:** The DeepFashion2 dataset can be downloaded from: https://github.com/switchablenorms/DeepFashion2 (accessed on 20 April 2022).

**Acknowledgments:** We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Quadro RTX 6000 GPU used for this research. This research has financially been supported by the Programme “FIL-Quota Incentivante” of University of Parma and co-sponsored by Fondazione Cariparma.

**Conflicts of Interest:** The authors declare no conflict of interest.

**References**

1. Ma, Y.; Jia, J.; Zhou, S.; Fu, J.; Liu, Y.; Tong, Z. Towards better understanding the clothing fashion styles: A multimodal deep learning approach. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017.
2. Jiang, S.; Wu, Y.; Fu, Y. Deep bi-directional cross-triplet embedding for cross-domain clothing retrieval. In Proceedings of the 24th ACM International Conference on Multimedia, Amsterdam, The Netherlands, 15–19 October 2016; pp. 52–56.
3. Li, X.; Wang, X.; He, X.; Chen, L.; Xiao, J.; Chua, T.S. Hierarchical fashion graph network for personalized outfit recommendation. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual, 25–30 July 2020; pp. 159–168.
4. Karras, T.; Laine, S.; Aittala, M.; Hellsten, J.; Lehtinen, J.; Aila, T. Analyzing and improving the image quality of stylegan. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 8110–8119.
5. Liu, Y.; Chen, W.; Liu, L.; Lew, M.S. Swapgan: A multistage generative approach for person-to-person fashion style transfer. *IEEE Trans. Multimed.* 2019, 21, 2209–2222. [CrossRef]
6. Zhu, S.; Urtasun, R.; Fidler, S.; Lin, D.; Change Loy, C. Be your own prada: Fashion synthesis with structural coherence. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 1680–1688.
7. Han, X.; Wu, Z.; Wu, Z.; Yu, R.; Davis, L.S. VITON: An Image-based Virtual Try-on Network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018.
8. Kim, B.K.; Kim, G.; Lee, S.Y. Style-Controlled Synthesis of Clothing Segments for Fashion Image Manipulation. *IEEE Trans. Multimed.* 2020, 22, 298–310. doi: 10.1109/TMM.2019.2929000. [CrossRef]
9. Jiang, S.; Li, J.; Fu, Y. Deep Learning for Fashion Style Generation. *IEEE Trans. Neural Netw. Learn. Syst.* 2021, 1–13. [CrossRef]
10. Huang, X.; Belongie, S. Arbitrary style transfer in real-time with adaptive instance normalization. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 1501–1510.
11. Choi, Y.; Uh, Y.; Yoo, J.; Ha, J.W. Stargan v2: Diverse image synthesis for multiple domains. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 8188–8197.
12. Ge, Y.; Zhang, R.; Wang, X.; Tang, X.; Luo, P. Deepfashion2: A versatile benchmark for detection, pose estimation, segmentation and re-identification of clothing images. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 5337–5345.
13. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. *Adv. Neural Inf. Process. Syst.* 2014, 27, 2672–2680.
14. Radford, A.; Metz, L.; Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv* 2015, arXiv:1511.06434.
15. Donahue, J.; Krähenbühl, P.; Darrell, T. Adversarial feature learning. *arXiv* 2016, arXiv:1605.09782.
16. Mirza, M.; Osindero, S. Conditional generative adversarial nets. *arXiv* 2014, arXiv:1411.1784.
17. Brock, A.; Donahue, J.; Simonyan, K. Large scale GAN training for high fidelity natural image synthesis. *arXiv* 2018, arXiv:1809.11086.

18. Karras, T.; Laine, S.; Aila, T. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, USA, 15–20 June 2019; pp. 4401–4410.

19. Zhang, H.; Xu, T.; Li, H.; Zhang, S.; Wang, X.; Huang, X.; Metaxas, D.N. StackGAN++: Realistic image synthesis with stacked generative adversarial networks. *IEEE Trans. Pattern Anal. Mach. Intell.* 2018, 41, 1947–1962. [CrossRef] [PubMed]

20. Dewi, C.; Chen, R.C.; Liu, Y.T.; Yu, H. Various generative adversarial networks model for synthetic prohibitory sign image generation. *Appl. Sci.* 2021, 11, 2913. [CrossRef]

21. Din, N.U.; Javed, K.; Bae, S.; Yi, J. A novel GAN-based network for unmasking of masked face. *IEEE Access* 2020, 8, 44276–44287. [CrossRef]

22. Isola, P.; Zhu, J.Y.; Zhou, T.; Efros, A.A. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, 21–26 July 2017; pp. 1125–1134.

23. Zhu, J.Y.; Park, T.; Isola, P.; Efros, A.A. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, Venice, Italy, 22–29 October 2017; pp. 2223–2232.

24. Choi, Y.; Choi, M.; Kim, M.; Ha, J.W.; Kim, S.; Choo, J. StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 18–22 June 2018; pp. 8789–8797.

25. Huang, X.; Liu, M.Y.; Belongie, S.; Kautz, J. Multimodal unsupervised image-to-image translation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, Munich, Germany, 8–14 September 2018; pp. 172–189.

26. Liu, M.Y.; Huang, X.; Mallya, A.; Karras, T.; Aila, T.; Lehtinen, J.; Kautz, J. Few-shot unsupervised image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision*, Seoul, Korea, 27 October–2 November 2019; pp. 10551–10560.

27. Gatys, L.A.; Ecker, A.S.; Bethge, M. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, 26–30 June 2016; pp. 2414–2423.

28. Johnson, J.; Alahi, A.; Fei-Fei, L. Perceptual losses for real-time style transfer and super-resolution. In *Proceedings of the European Conference on Computer Vision*, Amsterdam, The Netherlands, 11–14 October 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 694–711.

29. Ulyanov, D.; Lebedev, V.; Vedaldi, A.; Lempitsky, V.S. Texture networks: Feed-forward synthesis of textures and stylized images. In *Proceedings of the International Conference on Machine Learning (ICML)*, New York City, NY, USA, 19–24 June 2016; Volume 1, p. 4.

30. Li, C.; Wand, M. Combining markov random fields and convolutional neural networks for image synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, 26–30 June 2016; pp. 2479–2486.

31. Dumoulin, V.; Shlens, J.; Kudlur, M. A learned representation for artistic style. *arXiv* 2016, arXiv:1610.07629.

32. Gatys, L.A.; Ecker, A.S.; Bethge, M.; Hertzmann, A.; Shechtman, E. Controlling perceptual factors in neural style transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, 21–26 July 2017; pp. 3985–3993.

33. Li, Y.; Fang, C.; Yang, J.; Wang, Z.; Lu, X.; Yang, M.H. Universal style transfer via feature transforms. *Adv. Neural Inf. Process. Syst.* 2017, 30, 385–395.

34. Li, Y.; Wang, N.; Liu, J.; Hou, X. Demystifying neural style transfer. *arXiv* 2017, arXiv:1701.01036.

35. Zhang, Y.; Fang, C.; Wang, Y.; Wang, Z.; Lin, Z.; Fu, Y.; Yang, J. Multimodal style transfer via graph cuts. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, Seoul, Korea, 27 October–2 November 2019; pp. 5943–5951.

36. Chen, T.Q.; Schmidt, M. Fast patch-based style transfer of arbitrary style. *arXiv* 2016, arXiv:1612.04337.

37. Kiapour, M.H.; Yamaguchi, K.; Berg, A.C.; Berg, T.L. Hipster wars: Discovering elements of fashion styles. In *Proceedings of the IEEE Conference on Computer Vision*, Venice, Italy, 22–29 October 2017; pp. 2223–2232.

38. Johnson, J.; Alahi, A.; Fei-Fei, L. Perceptual losses for real-time style transfer and super-resolution. In *Proceedings of the European Conference on Computer Vision*, Amsterdam, The Netherlands, 11–14 October 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 694–711.

39. Ulyanov, D.; Lebedev, V.; Vedaldi, A.; Lempitsky, V.S. Texture networks: Feed-forward synthesis of textures and stylized images. In *Proceedings of the International Conference on Machine Learning (ICML)*, New York City, NY, USA, 19–24 June 2016; Volume 1, p. 4.

40. Li, C.; Wand, M. Combining markov random fields and convolutional neural networks for image synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, 26–30 June 2016; pp. 2479–2486.

41. Dumoulin, V.; Shlens, J.; Kudlur, M. A learned representation for artistic style. *arXiv* 2016, arXiv:1610.07629.

42. Gatys, L.A.; Ecker, A.S.; Bethge, M.; Hertzmann, A.; Shechtman, E. Controlling perceptual factors in neural style transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, 21–26 July 2017; pp. 3985–3993.

43. Li, Y.; Fang, C.; Yang, J.; Wang, Z.; Lu, X.; Yang, M.H. Universal style transfer via feature transforms. *Adv. Neural Inf. Process. Syst.* 2017, 30, 385–395.
44. Yang, X.; Ma, Y.; Liao, L.; Wang, M.; Chua, T.S. Transnfcm: Translation-based neural fashion compatibility modeling. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 403–410.
45. Becattini, F.; Song, X.; Baecchi, C.; Fang, S.T.; Ferrari, C.; Nie, L.; Del Bimbo, A. PLM-IPE: A Pixel-Landmark Mutual Enhanced Framework for Implicit Preference Estimation. In ACM Multimedia Asia; 2021; Association for Computing Machinery; New York; NY; USA; Article 42; pp. 1–5.
46. De Divitiis, L.; Becattini, F.; Baecchi, C.; Bimbo, A.D. Disentangling Features for Fashion Recommendation. ACM Trans. Multimed. Comput. Commun. Appl. (TOMM) 2022. [CrossRef]
47. Divitiis, L.D.; Becattini, F.; Baecchi, C.; Bimbo, A.D. Garment recommendation with memory augmented neural networks. In Proceedings of the International Conference on Pattern Recognition, Virtual, 10–15 January 2021; Springer: Berlin/Heidelberg, Germany, 2021; pp. 282–295.
48. Yoo, D.; Kim, N.; Park, S.; Paek, A.S.; Kweon, I.S. Pixel-level domain transfer. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 517–532.
49. Lassner, C.; Pons-Moll, G.; Gehler, P.V. A generative model of people in clothing. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 853–862.
50. Jetchev, N.; Bergmann, U. The conditional analogy gan: Swapping fashion articles on people images. In Proceedings of the IEEE International Conference on Computer Vision Workshops, Venice, Italy, 22–29 October 2017; pp. 2287–2292.
51. Raffiee, A.H.; Sollami, M. Garmentgan: Photo-realistic adversarial fashion transfer. In Proceedings of the 2020 25th International Conference on Pattern Recognition (ICPR), Milan, Italy, 10–15 January 2021; pp. 3923–3930.
52. Lewis, K.M.; Varadharajan, S.; Kemelmacher-Shlizerman, I. Tryongan: Body-aware try-on via layered interpolation. ACM Trans. Graph. (TOG) 2021, 40, 1–10. [CrossRef]
53. Li, P.; Xu, Y.; Wei, Y.; Yang, Y. Self-correction for human parsing. IEEE Trans. Pattern Anal. Mach. Intell. 2020, 44, 3260–3271. [CrossRef]
54. Zhang, R.; Isola, P.; Efros, A.A.; Shechtman, E.; Wang, O. The unreasonable effectiveness of deep features as a perceptual metric. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 586–595.
55. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. arXiv 2014, arXiv:1409.1556.
56. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. arXiv 2014, arXiv:1412.6980.
57. Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; Hochreiter, S. Gans trained by a two time-scale update rule converge to a local nash equilibrium. Adv. Neural Inf. Process. Syst. 2017, 30, 6629–6640.