



Style Accessory Occlusion using CGAN with Paired Data

Sujith Gunjur Umapathy¹, Alexander I. Iliev^{1,2}(✉)

¹ SRH University Berlin, Charlottenburg, Germany

² Institute of Mathematics and Informatics, Bulgarian Academy of Sciences, Sofia, Bulgaria
sujithgunjur.umapathy@stud.srh-campus-berlin.de;
ailiev@berkeley.edu

Abstract. Virtual try-on (VTO) is an exciting and captivating concept where end users or customers can gain a near-realistic experience of the product/ item they are interested in. In the field of VTO, Augmented Reality (AR) and Virtual Reality (VR) actively contribute. However, these technologies require sophisticated hardware equipment like a VR Glass /VR Headgear or computationally capable smartphones to provide a good experience with AR. To solve this issue of requiring expensive hardware, Machine Learning (ML) and Artificial Intelligence (AI) have contributed to research and applications in VTO. To this end, Generative Modelling can be considered, especially for VTO on an e-commerce platform. Generative models are volatile. To solve this instability issue and to create a unique experience with generative models for the VTO domain, paired-wise data is used to occlude a style accessory like Hat/ Cap on portrait images.

Keywords: Generative Modelling, CGAN, Pix2Pix.

1 Motivation

E-commerce platforms are designed to host a variety of products ranging from fashion to household appliances. It is beneficial for both parties, the seller, and the buyer. Amazon CEO Jeff Bezos, among others, sees a co-relationship, existing with customers returning to their choice of e-commerce platforms (Richard, 2022) ^[2]. They are:

1. Eye-catching elements
2. Simple Design
3. Product offers and promotions
4. Security of user data and more

Nearly 76 out of 100 customers (Richard, 2022) ^[2] tend to abandon an e-commerce platform if any of the above features are misaligned with the actual expectation (See Fig 1.) A report from Jungle Scout ® ¹ (Boice, 2021) ^[1] indicates that nearly 73% of the consumers in the U.S. prefer to shop online in the future. E-commerce platforms are continuously innovating to improve the customer experience and simplify purchases. This shift in dynamics is mainly due to the economy created during the pandemic,

¹ <https://www.junglescout.com>

which altered the spending habits of the consumers. The above report indicates that nearly 37% of consumers among the curated group preferred to shop online mainly due to convenience. Another group of consumers, i.e., almost 36%, indicated that ease of purchase of a specific product motivated them to return to their chosen e-commerce platform.

This fuels every e-commerce giant to stay on top of the game and churn the innovative cycle to generate new ideas and increase convenience and eye-catching elements within their platform

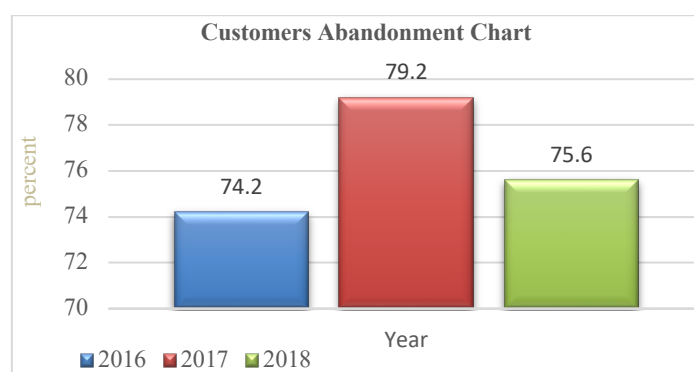


Fig. 1. Customer Abandonment measured by Big Commerce (R) ²

1.1 Problem

Though online shopping enhances the ease and convenience of product purchases, one should not forget that the investments are made by browsing through an electronic screen. Further, customers who are unhappy with their purchase tend to return the product. An article from Shopify® ³ (Dopson, n.d.) ^[3], a leading online retailer, states that product returns majorly hurt the profit margins, hindering the conversion rate, finally hurting the business. The report also indicates that the National Retail Federation ⁴ (NRF) accounts for nearly \$400 billion (National Retail Federation, n.d.) ^[4] in lost sales for U.S. retailers alone.

To understand the reason for the return, Shopify investigated further to reduce the negative business impact, which is accompanied by the product returns. The report indicates that (Dopson, n.d.) ^[3], on average, 12.2% of apparel and 4.3% of beauty products (See Table 1.) are returned by consumers. Further, a customer's return survey conducted by ReBound® in 2020 (The Returning Conundrum, n.d.) ^[5] indicated that nearly 5-15% of the items are usually returned by almost 24.9% of the consumers (See Fig 2.).

² <https://www.bigcommerce.com>

³ <https://www.shopify.com>

⁴ <https://nrf.com>

Among this distribution, customers mainly return due to either change of mind, an unsatisfied style, or the product is not as described (See Fig 3).

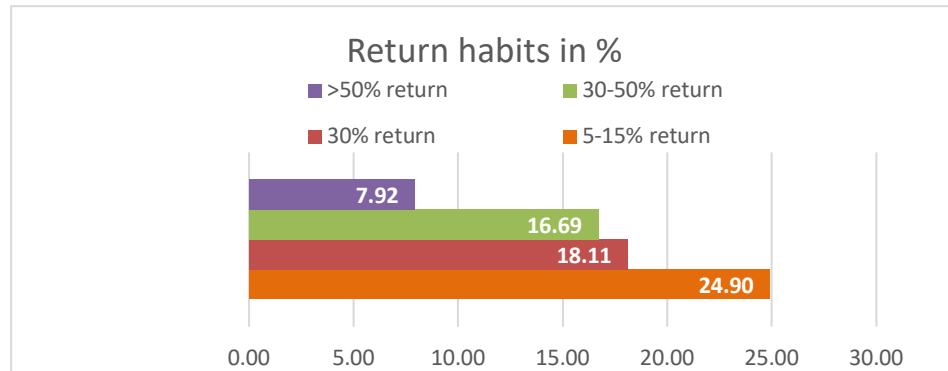


Fig 1. Customer return habits (The Returning Conundrum,n.d.) ^[5]

Fig 3. indicates that mainly, “Change of Mind” and “Unsatisfied Style” causes product return problems. Innovation in this domain especially allows customers to make willful decisions.

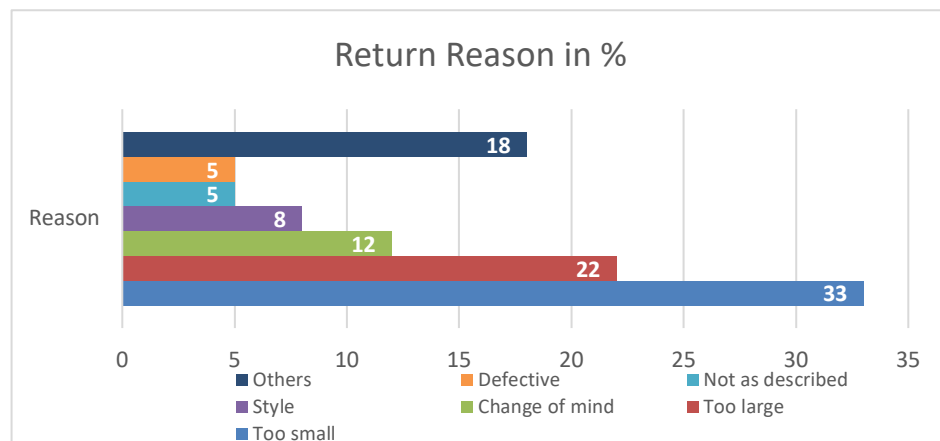


Fig 2. The reason why customers return their products (The Returning Conundrum, n.d.) ^[5]

Table 1. Return Rate by Category (Dopson, n.d.) ^[3]

Retail Category	Return Rate
Apparel	12.2 %
Auto parts	19.4 %
Beauty	4.3 %
Department stores	11.4 %
Housewares	11.5 %

Under the fashion category, it's important to give online customers an immersive experience and ease their purchase decision. This, in turn, benefits the platform owners with higher customer retention and sales and reduces the return of products. Further, using generative models, the platform owners need not invest highly in AR/VR technology or their development cycle but rather serve a GAN model, which can superimpose style accessories like hats or caps. With the use of the GAN model, the generation of results is faster, which in turn adds to a better CX.

1.2 Relevance

Personalization is one of the critical factors in purchasing decisions (Nedunchezian, 2020) ^[6]. In the fashion category, products and softlines grab a customer's attention when product images depict the true nature of that clothing product. To capture this, softline products are shot in a studio environment to produce high-quality results. This provokes creativity in sellers. Creativity here translates to creating product portfolios with,

- A. Day to Day accessories
- B. Natural scene-setting
- C. Fabric durability and innovative design
- D. Color variations
- E. Skin tone variations

However, this scenario is limited to high-profile sellers as it involves higher capital investment and risk. These sellers often create virtual experiences to promote personalization. For example, Loreal®, a famous hair color brand, has a virtual hair color try-on experience⁵ on their web portal. These personalization features empower customers to make the right decision, increasing the number of satisfied customers and increasing customer retention and sales. Small businesses and startups often do not have the resources to create a dynamic and attractive product portfolio and personalization features that can include the targeted Softline and different fashion accessories. This is due to the complex attributes of soft lines, i.e., soft line patterns, textures, vibrant colors, etc.

2 Introduction

Generative Adversarial Networks are the machine learning task of learning from the underlying pattern in such a way that model learns from the regularities and replicates the underlying representation/ image based on the training set. To this end, GANs frame the problem as a supervised learning model with two sub-models: the generator and the discriminator. The generator learns to generate the images, and the discriminator is trained to classify an image as an image.

In this training network, the generator tries to generate examples close to the actual training distributions and learns to minimize the error by fooling the discriminator. However, the discriminator is trained to identify images as generated or originated from the training set and, in turn, maximizes the likelihood of identifying the image as real

⁵ <https://www.loreal-paris.co.uk/virtual-try-on>

or fake. This adversarial nature is governed by the adversarial loss (Goodfellow, 2014) [7] and it is calculated as:

$$\min \max \text{ loss} = E_x[\log D(x)] + E_z[1 - D(G(z))]$$

where,

E_x = Expected values on a real data instance

$D(x)$ = Discriminators estimate that the real data instance x is real

$G(z)$ = Generators output for a given latent vector z

$D(G(z))$ = Discriminators estimate that the fake data instance is real

GANs can generate hyper-realistic images, as seen in Fig 4. This is mainly due to the adversarial nature of this network, where the adversarial loss can help the Generator to improve progressively with each step. Once the generator starts producing realistic images, the generator can be separated from the network and used to generate hyper-realistic images. Unconditioned generative models can create images like the training distribution using random noise. However, this deviates from the motivation. Further conditioning on GANs is required to generate images with specific requirements,

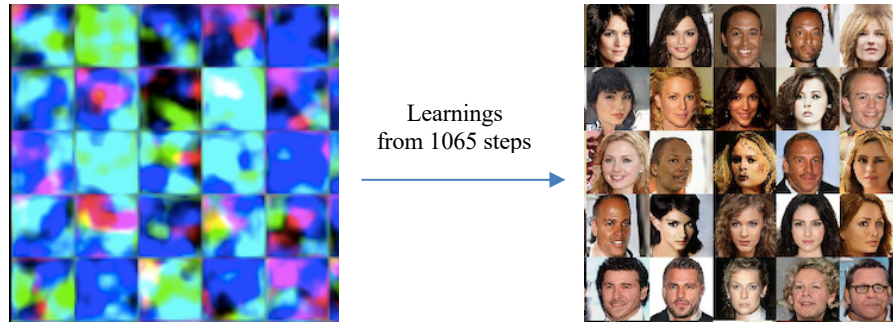


Fig 4. Style GAN ADA image generation after 1065 steps, trained on CelebA ⁶ dataset

2.1 Conditional GAN (CGAN)

Conditional GANs are novel generative models that are a conditioned version of generative adversarial networks constructed by feeding a conditional variable y (Mirza, 2014) [8]. In unconditional GANs, the GAN generations are random, and they can generate different modes and would not have any control over the generation. However, in Conditional GANs, it is possible to direct the generation process toward a particular mode. Such conditioning involves a variable in the form of a class label either from the same or a different modality.

The objective function (Mirza, 2014) [8] is as follows:

$$\min \max V(D, G) = E_{x \sim p_{data}(x)}[\log D(x|y)] + E_{z \sim p_z(z)}[\log(1 - D(G(z|y)))].$$

⁶ <https://mmlab.ie.cuhk.edu.hk/projects/CelebA.html>

where,

Y – extra information or the conditional variable

$P_z(z)$ – generator input noise

X – input data

D – discriminator

G – generator

Z – latent vector

The aspect of conditionality is highly relevant to the proposed problem. The primary motivation to use CGANs is that the training process can be preconditioned on a variable that can control the outcome of the generative model. In this paper, the training process is conditioned mainly on the specific type of hat/ caps or any other style accessory. With this precondition, the generative model is forced to generate images of a particular category of hat/caps. The discriminator, on the other hand, would classify images as fake 1) Suppose the pair of generated images and its class labels are not the same, and 2) Suppose the generated image can be easily classified as fake. Further, this classification helps in the training process through backpropagation, where the generator learns to create images that can 1. Fool the generator, and 2. The generated images are conditioned to the class label. Further, conditioning can also be asserted on the image's domain as well. This is facilitated through the popular Pix2Pix architecture.

2.2 Pix2Pix Architecture

Pix2Pix, or the image-to-image translation model, is the crux of our proposed network architecture. This architecture is used to solve the problem statement of being able to generate/occlude style accessories on the input image.

A Pix2Pix model is a conditional generative model, where the output image is conditioned on an input image. The discriminator network is provided with both the generated image and the real image. The task of the discriminator is to determine whether the generated image is a possible transformation of the real image. The generator is required to generate images that resemble the images from the training set. To this end, the generator network is trained in an adversarial nature, primarily with an L1 loss on the whole picture. The L1 loss is measured on the real image and the generated image.

The objective function for this conditional generator is given by the following equation (Isola, 2016) ^[9].

$$G^* = \arg_{\min \max} L_c GAN(G, D) + \lambda_L L1(G)$$

where,

- $LGAN(G, D) = E_y [\log D(x, y)] + E_{x, z} [\log(1 - D(G(x, z)))]$ ⁷
- $L_{L1}(G) = E_{x, y, z} [\|y - G(x, z)\|_1]$

⁷ This equation represents the conditional variant, which observes the input image 'x.'

- G- Generator
- D- Discriminator
- G*- Adversarial loss where G tries to minimize the objective and D tries to maximize the same
- Z – Gaussian noise
- X- input image from the domain.'
- Y – target image
- L_{L1} – L1 loss
- LGAN- conditional GAN

The generator architecture is mainly comprised of the U-Net-like architecture, where the input is passed through progressive levels of the layer to downsample the input image. All the information needs to pass through these layers and the bottleneck. At this point, the process is reversed by the up-sampling layer. To allow the network to maintain the fine-grained information through the bottleneck, skip connections are used that can preserve the high-level information. Skip connections usually exist between layer 'I' and layer 'N-I.'

The discriminator architecture is this network relies on modeling two structures. One is the high-frequency structure, and the other is the low-frequency structure. The latter is taken care of by the L1 or the L2 loss. Though L1 and L2 produce blurry results (Isola, 2016)^[9], the low-frequency structure is well maintained by the L1 loss function in this architecture. To preserve the high-frequency structure, close attention is paid only to the portion or a patch of the whole image at a time. This mechanism is called PatchGAN, as it penalizes the whole structure on a patch-level basis. In this architecture, the discriminator classifies $N \times N$ patches as accurate or as an image generated by the generator

With this network architecture, paired-wise data plays an important role during the training process. In our paper, we approach occlude style accessories such as a hat or a cap on portrait images. Creating paired-wise data with real humans is tedious and time demanding. Hence, we addressed this issue by creating a custom dataset using 3D avatars.

3 Dataset

Pix2Pix or Image-to-Image architecture mentioned in the previous section remains the basis of the implementation. However, it is also to note that the model architecture solely depends on pair-wise data. In our proposal, pair-wise data translates to a pair of images where source image A is a portrait of a human not wearing a style accessory like a hat/cap, and the target image B is the same portrait of a human wearing a style accessory like a hat/cap. Further requirements are listed below:

1. Must be of the same color space
2. Must have humans in the same pose
3. Human faces must have the same orientation

4. The spatial and temporal features of both images must remain the same. This will prevent the model from drifting from the problem statement
5. Images should have a 3D depth to create a realistic image generation

This is a bottleneck in the implementation step, as there are no open-source datasets that can fulfill all the requirements mentioned above. To this end, a dataset was created keeping the above-mentioned requirements in mind. Conforming to these requirements allows the model to train as expected and generate an image with a style accessory like a hat or a cap. Further, a 3D avatar rendering and interactive portal, **Ready Player Me**⁸ is used. Avatars have become essential for creating and playing graphical games, AR/VR apps, and virtual events. The generated avatars can be used on mobile platforms like Android and iOS and game engines like Unity and Unreal.

3.1 Dataset Composition

To ensure that the GAN model is not biased, the dataset is composed by including different gender, ethnicity, hairstyles, beard styles, and other accessories w.r.t to sunglasses/eyewear. Dataset composition can be seen in Fig 5.

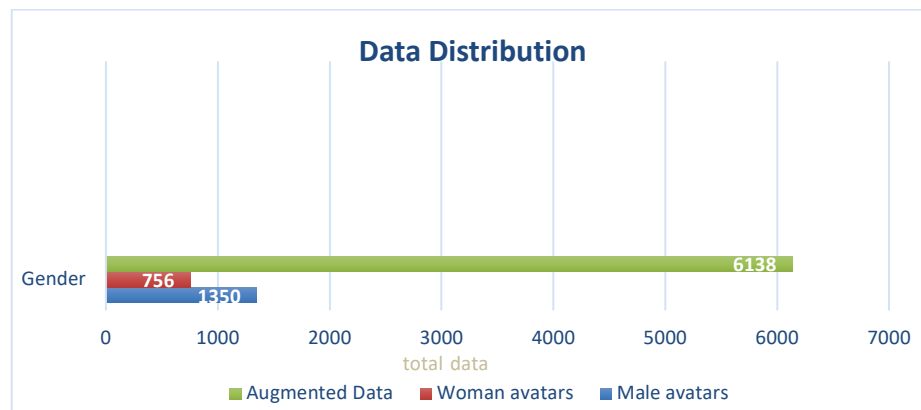


Fig 5. Data distribution in our dataset

A dataset with the following composition was generated. Each image is a 3-dimensional RGB color-spaced image with a size of 400,400. The augmented data is generated using alumentation⁹, an open-source library for creating augmented images. The images were augmented mainly with x-axis shift, rotation $\pm 20^\circ$, and scale $\pm 10\%$

Different skin tones and gender. The dataset comprises avatars of different ethnicity, namely Caucasian males, Tanned males, and Asian-origin males (See Fig 6). Further, it includes Asian females and females with blonde hair as well.

⁸ <https://readyplayer.me>

⁹ <https://alumentations.ai>



Fig 6. Avatars with different racial ethnicity

Different hats and caps. To improve the learning from different hat shapes and colors (See Fig 7). The dataset also includes other types of hats/caps. And they are 1) Cap, 2) Beanie, 3) Boat cap, 4) Cowboy hat Type-1, and 5) Cowboy hat Type-2



Fig 7. Avatars with different head gears

Different eyewear. To improve the diversity of training images, avatars wearing different eyewear, such as spectacles type-1, aviators, wayfarers, and spectacles type-2 are included. Further variations can be seen in woman's eyewear, mainly with color and frame shapes (See Fig 8).



Fig 8. Avatars with different eye wear

Different looks. Further, to enhance the diversity of image generations, the style factors of the avatars are also varied. For male avatars, different types of beards and facial hair are included. For female avatars, different makeup looks with varying eye shadows are included (See Fig 9).



Fig 9. Avatars with different beard types

Different head rotation. To provide a better visual perception of the worn accessory, different head rotations for every character with all possible permutations of the hair

type, looks, and caps are included. With this, the dataset size increases and improves the diversity of the images on which the GAN network can be trained (See Fig 10).



Fig 10. Avatars with different head rotation

These styles add up to the total sample set as shown in Fig 11., which can be used to train a generator model. However, in the training process, the standard TensorFlow Dataset objects are created from the API '*tf.data.Dataset.list_files()*'. This API expects the training or validation images to be in their separate folders.



Fig 11. Style variation in our dataset

However, in the model architecture mentioned in the previous section, it is required to feed a source image, source cap/hat, and the target image. To this end, the three images will be concatenated into one, and later separate into these images in the preprocessing stage. Sample images from the real and augmented dataset can be seen in Fig 12 and Fig 13, respectively.



Fig 12. Non-augmented paired data



Fig 13. Augmented paired data

3.2 Dataset Validation

To ensure that this dataset conforms to the actual human attributes w.r.t hair, eyes, head, etc., it is essential to qualitatively and quantitatively measure the results generated using this dataset.

The Fréchet Inception Distance (FID) score, which assesses the quality of the GAN-generated image, is nearly 20 from our training dataset. Further elaboration can be viewed in Table 2.

To further assess the useability of this dataset, a pretrained semantic segmentation model was run on our dataset. The hypothesis is that if the semantic parser can identify facial attributes on our dataset, it is safe to assume that our dataset is composed of human-like objects. The parser successfully identified the face's semantics without pre-processing the input image.

Table 2. FID comparison against real-world data

Dataset	Size of the dataset	FID (lower the better)
CIFAR10 ¹⁰	30000	2.42
FFHQ ¹¹	70000	2.84
Our dataset	2106 (true data)	~19.34

An example result can be seen in Fig 14. The pretrained model is provided by the CelebAMask HQ dataset provider ¹². The per-pixel accuracy on CelebAMask HQ dataset is 93.42. The pretrained model ¹³ is trained on a dataset containing 30000 high-resolution images with the face semantic segmentation Ground Truth (GT).

¹⁰ <https://www.cs.toronto.edu/~kriz/cifar.html>

¹¹ <https://github.com/NVlabs/ffhq-dataset>

¹² https://github.com/switchablenorms/CelebAMask-HQ/tree/master/face_parsing

¹³ <https://tinyurl.com/2cjas49s>



Fig 14. Input image and segmentation model output (L-R)

4 Implementation

This section mainly discusses the implementation details involved in solving the problem statement mentioned earlier.

To this end, the focus of this implementation is to solve the problem statement which the e-commerce platforms mainly encounter, i.e., the customers returning products purchased on an online site claiming not to be as described, or the customers were dissatisfied with the product style. This can impact sales significantly for any online platform. To reduce this, there is a need to develop VTO solutions that are simple and easy to onboard for any online platform. To this end, the implementation focuses on occluding style accessories like caps or hats onto custom avatars. However, the texture and color of these accessories also need to be maintained. This would allow customers to visualize a product/style accessory before making a purchase.

Further, the hypothesis is that the Return of Products(ROP) reduces, thus lowering the impact on sales.

This implementation expects to train a generative model that can translate the image from the no-hat domain to the hat domain. However, since the customer expects the chosen cap to be occluded, the generative image translation model must also be conditioned on the given hat.

4.1 Architecture

The model architecture follows the line of Pix2Pix architecture. However, Pix2Pix is conditioned only on the target image. But that conditioning will not suffice the problem statement that is being approached. It also needs to condition the input style accessory to transfer the style, texture, and color onto the source image, to finally generate an output image.

The main components in this architecture are the generator and the discriminator. The generator and discriminator take two inputs of shape $(h,w,3)$ as shown in Fig 15. As inputs, a source image to occlude the style accessory and a source style accessory (A hat/cap of a specific color and texture) is provided. Using this input, the generator would produce an image like the images in the training distribution.

The task of the discriminator is to then receive the image from the source distribution along with the style accessory present in the target image. Along with these, the discriminator model requires the generated image as input along with the chosen hat or cap. The discriminator is allowed to classify real vs. fake as per the following scenario:

1. *Real Image* – The style accessory matches the occluded style accessory, and the discriminator cannot distinguish the image as fake.
2. *Fake Image* – The style accessory does not match the style accessory in the generated image
3. *Fake Image* – The style accessory does match the style accessory in the generated image. However, the generator model failed to fool the discriminator. This information is computed and backpropagated through the network.

It is important also to note that, the generative model needs to perfectly regenerate the source image and the texture/ color/ style of the given hat or cap. This mechanism is crucial to creating a better and more helpful visualization for the customers of any e-commerce platform.

To emphasize this mechanism, generator weights are incorporated as shown in Fig 10. These generator weights emphasize the generator to produce images exactly like the source image using the L1 loss. L2 loss can also be used in the generation process, however. However, the results are blurry. This phenomenon is called out in several works of literature (Isola, 2016; Zhao et al., 2017; Salem et al., 2018)^[9,10,11]. However, the overall prediction improves with L2 loss. Hence L2 is weighted in half when compared to the L1 loss.

In this architecture, the L1 loss is weighed two orders of magnitude more than that of the adversarial loss. In this manner, the image generation process is conditioned to strictly conform to the source image and a given style accessory.

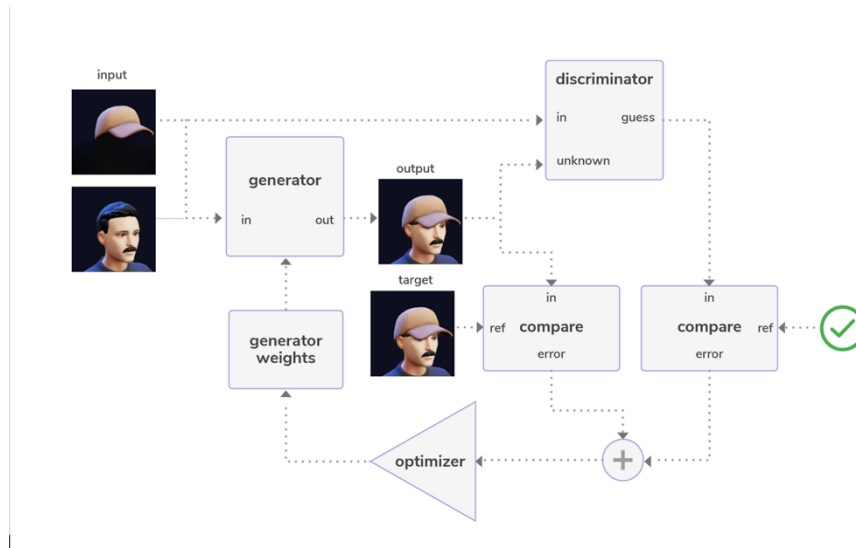


Fig 15. The architecture of the proposed solution

4.2 Generator Objective Function

The generator objective function is shown in the below equation. However, this equation slightly modifies the pix2pix equation. The generator network can be seen in the Appendix.

$$\text{HatGAN} = \arg_{\min \max} L_c \text{GAN}(G, D) + R * L1(G * 0.6) + R/2 * L2(G * 0.6)$$

where,

- $LGAN(G, D) = E_y [\log D(x, s, y)] + E_{x, s, z} [\log (1 - D(x, G(x, s, z)))]$
 - x, z is of 3 dimensions, RGB color space image.
- $L_{L1}(G) = E_{x, y, s, z} [\|y - G(x, s, z)\|_1]$
- G- Generator
- D- Discriminator
- HatGAN- Adversarial loss where G tries to minimize the objective and D tries to maximize the same
- Z – Gaussian noise
- X- input image from the domain
- Y – target image
- L_{L1} – L1 loss
- L_2 – L2 loss
- R – Weight factor
- LGAN- conditional GAN
- $L1(G*0.6)$ – Signifies the L1 loss computed over 60% of the image from the top (This inclusion is to add more weight and introduce bias to the style accessory and the face. 60% of the composition from the top will include the hat/cap along with the face)

4.3 Discriminator Objective Function

The below equation gives the discriminator's objective function. This is a simple equation with the binary cross entropy loss for the classification of real vs. fake images. The discriminator network can be seen in the Appendix.

$$BC = \frac{1}{N} \sum_1^N - (y * \log(P) + (1 - y) * \log(1 - p))$$

$$\text{HatD} = BC(\text{Real}, 1's) + BC(\text{Fakes}, 0's)$$

where,

- BC – Binary Cross entropy
- N – Number of classes
- Y – Label
- P – Probability of positive classes
- 1-P – Probability of negative classes
- HatD – HatGAN discriminator

- Real – Image from the actual distribution
- Fake – Image from the generator
- 1's – An nD array of 1's
- 0's – An nD array of 0's
- The primary goal of this objective function is to maximize the discriminator's output by predicting the correct class label for real and generated images.

4.4 Model inference

Training happens in two different networks, in a step iteration. The two networks are namely the generator network and the discriminator network. The generator and the discriminator include a series of convolutional layers. The generator is composed using the U-Net architecture, which involves a) downsampling the images to lower dimensions using Conv2D, b) upsampling images from lower dimensions to higher dimensions, and c) finally concatenating the learnings from higher and lower dimensionality using the skip connections. Further, dropout layers are also included in the first three up-sampling layers. The discriminator is a simple convolution layer that samples the image from (256, 256, 9) to an embedding of (30,30,1). The last two layers in the discriminator network have a stride of 1. However, the remaining layers have a stride of two.

In this training process, the model is fit for a predefined number of steps. In the experimentations, this training step ranges between 100K and 200K. Source images and hat are concatenated for each step and fed into the generator network. This is further concatenated with the target image in the generator network. For each step, the generator generates an image and computes the binary cross entropy loss against a matrix of one's. This is further combined with L1 and L2 loss with different weights. Total generator loss is the summation of the binary cross-entropy loss along with L1 and L2 loss.

Further, after calculating the generator loss, the discriminator loss is computed as the summation of binary cross entropy loss of real image classification and fake image classification. After calculating both the loss functions, gradients are calculated and backpropagates. The network alters the weights accordingly and proceeds with the next step

5 Results

The following training configurations (See Table 3) yielded high fidelity and diversity, during the training process. Few results from the training set can be seen in Fig 16,17. Discriminator Loss, Generator Loss and L1 loss curves can be seen from Fig 19, 20, 21.

Table 3. Training specification

Image Ordering	Randomized order from the dataset with male and female avatars
Training data size	11000 images (referred to as the Golden dataset)
Test data	1411 images

Augmented	Yes
Male and female avatars	Yes
Information	Conditioned on the style accessory like hat/caps
Train steps	200K
Optimizer	Adam with LR 0.0001 for generator and 0.0004 for discriminator
Loss	Adversarial Loss + L1 loss + L2 loss. Error is calculated on 60% of the image alone

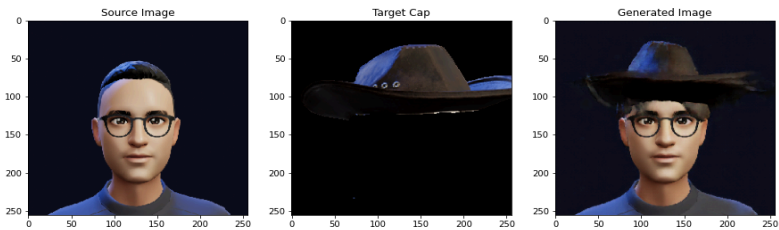


Fig 16. Example 1 from the final training configuration, from the training set

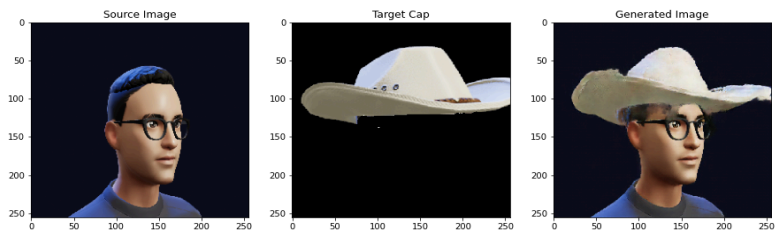


Fig 17. Example 2 from the final training configuration, from the training set

Cap Swap Test. In this test, two images are randomly chosen. The hats of these two pairs are swapped with each other. For example, below a single image generation pair is shown. Here the ‘changed’ section represents the hat style that was changed from the adjacent image pair (See Fig 18).

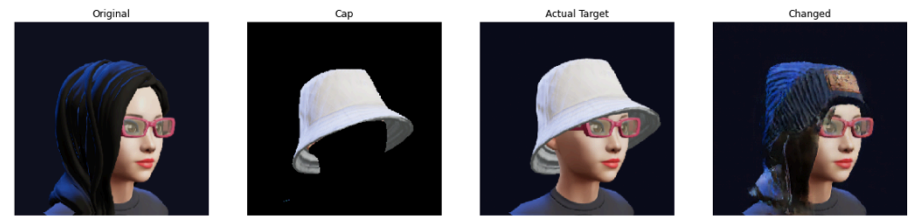


Fig 18. Example output from the cap swap test, from the training set

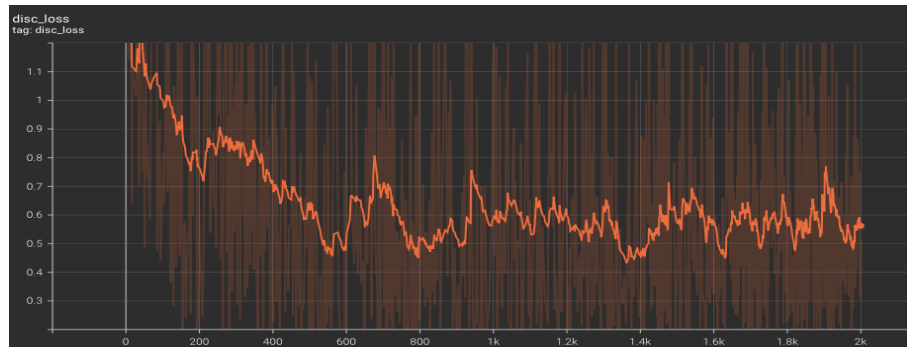


Fig 19. Discriminator Loss, Smoothed value = 0.5618, Actual Value = 0.3029

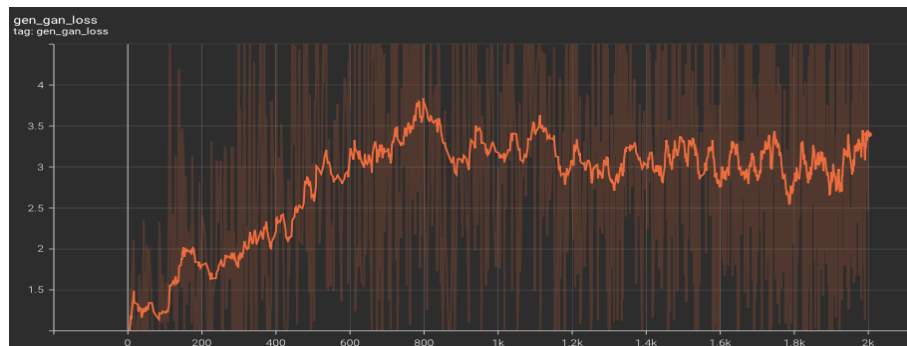


Fig 20. Generator Loss, Smoothed value = 3.39, Actual Value = 2.46

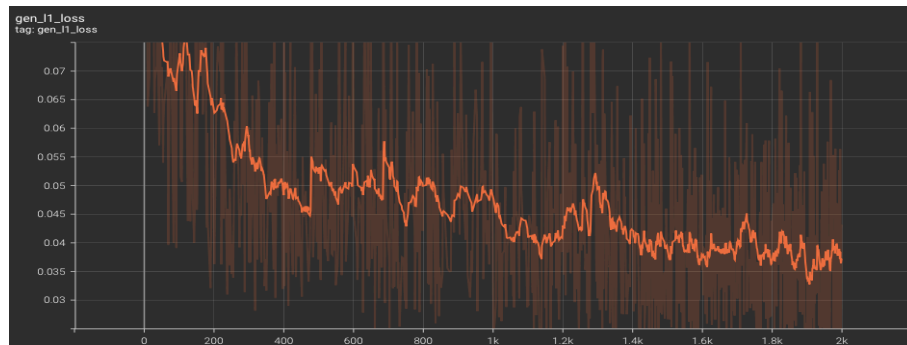


Fig 21. L1 loss curves, Smoothed value = 0.037, Actual Value = 0.043

6 Conclusion

In this paper, a major problem that the e-commerce platforms are currently facing is being solved i.e., the customer return habits. It is a common scenario, for customers to try a product and return them when it doesn't meet their expectations. In this era, all the e-commerce platforms frame a generous return policy mainly to improve customer

trust and customer retention. However, this indeed comes with side effects. From another online report (Nosto, n.d.), nearly **50%** of customers don't want to pay for product returns. This directly translates that; the online platforms need to bare the cost. E-commerce platforms tend to face a capital loss due to the return policy. To compensate for this, either the customer or the sellers could be overburdened at a later point.

To provide a solution to this, in this paper, a generative modeling approach is proposed wherein the customers are allowed to upload their image/selfie along with an accessory like a hat that they would like to try on. A trained generator network generates an image with an accessory occluded onto the customer image, with the right spatial and temporal specification. This architecture can be extended to any other style accessory and will allow the training of a generative model. The proposed architecture is like the pix2pix GAN network; however, the architecture varies in multiple folds. A major change in this architecture is that the generator and the discriminator are conditioned on two images, the source, and the accessory. To further strengthen our hypothesis of being able to occlude a style accessory using generative models, the model's performance is evaluated on the test dataset of ~1400 images with a few images outside of the training dataset distribution. The model continued to generate consistent images along with the style accessory, in the right spatial configuration. Further, this generation technique can be used on human data and can largely benefit e-commerce platforms, by providing their customers with a VTO experience in the soft lines category.

References

- [1] Boice, M. (2021, June 17). *Top Reasons Consumers Shop Online - Why Online Shopping is Popular*. Jungle Scout. <https://www.junglescout.com/blog/reasons-consumers-shop-online/>
- [2] Richard, C. (2022, March 9). *Ecommerce UX: What It Takes To Create The Best User Experience For Your Online Store*. The BigCommerce Blog. <https://www.bigcommerce.com/blog/ecommerce-ux/#ecommerce-ux-drives-conversion>
- [3] Dopson, E. (n.d.). *How to Handle and Prevent Ecommerce Returns for Maximum Profitability*. Shopify Plus. <https://www.shopify.com/enterprise/ecommerce-returns>
- [4] National Retail Federation. (n.d.). *Customer Returns in the Retail Industry*. NRF. <https://nrf.com/research/customer-returns-retail-industry>
- [5] The returning conundrum. (n.d.). IMGR. https://f.hubspotusercontent10.net/hubfs/2182667/The%20Returning%20Conundrum.pdf?utm_campaign=IMRG%20Report%20-%20July%202021&utm_medium=email&_hsmt=140696218&_hsenc=p2ANqtz-8K4OK-9cu2w0SgS3F7Hxhq9BfXJAwV9iAP11ROVu-_9SJt3PfMrvfHiKB03e0g880hjiNyvU0JH2BwNbpUqIJFCq-f5w&utm_content=140696218&utm_source=hs_automation
- [6] Nedunchezian, B. S. A. V. R. (2020, February 22). *Factors Influencing Customers Buying Decision towards Shopping Online and Offline with Reference to Coimbatore City*. Factors Influencing Customers Buying Decision towards Shopping Online. <https://www.abacademies.org/articles/factors-influencing-customers-buying-decision-towards-shopping-online-and-offline-with-reference-to-coimbatore-city-8973.html>

- [7] Goodfellow, I. J. (2014, June 10). *Generative Adversarial Networks*. arXiv.Org. <https://arxiv.org/abs/1406.2661>
 - [8] Mirza, M. (2014, November 6). *Conditional Generative Adversarial Nets*. arXiv.Org. <https://arxiv.org/abs/1411.1784>
 - [9] Isola, P. (2016, November 21). *Image-to-Image Translation with Conditional Adversarial Networks*. arXiv.Org. <https://arxiv.org/abs/1611.07004>
 - [10] Salem, N. M., Mahdi, H. M. K., & Abbas, H. (2018). Semantic Image Inpainting Using Self-Learning Encoder-Decoder and Adversarial Loss. *2018 13th International Conference on Computer Engineering and Systems (ICCES)*. <https://doi.org/10.1109/icces.2018.8639258>
 - [11] Zhao, H., Gallo, O., Frosio, I., & Kautz, J. (2017). Loss Functions for Image Restoration With Neural Networks. *IEEE Transactions on Computational*
- Nosto. (n.d.). *Ecommerce Return Rate Statistics in 2021: Causes and Best Practices*. <https://www.nosto.com/ecommerce-statistics/return-rate/>