International Journal of Computational and Experimental Science and ENgineering (IJCESEN)

Vol. 11-No.3 (2025) pp. 4545-4554
http://www.ijcesen.com

ISSN: 2149-9144

Research Article

An Optimized Generative Adversarial Network Model for Virtual Try-On: Enhancing Image Realism with Particle Swarm Optimization Algorithm

Aqeel Salam Kashkool[1]*, Amir Lakizadeh[2]

[1] Department of Computer Engineering and Information Technology, University of Qom, Qom, Iran
*** Corresponding Author Email:** actionpublication15@gmail.com - **ORCID:** 0000-0002-5247-7830

[2] Department of Computer Engineering and Information Technology, University of Qom, Qom, Iran
Email: ami2r@gmail.com - **ORCID**: 0000-0002-5247-7820

Article Info:

DOI: 10.22399/ijcesen.3049
Received : 21 March 2025
Accepted : 19 June 2025

Keywords

Deep Learning
GAN
Virtual Try-n
Particle Swarm Optimization (PSO)

Abstract:

Traditional methods of Virtual try-on of clothes wearable products face challenges such as high cost, time-consuming and lack of exact matching; Therefore, nowadays, the use of virtual testing has become important due to its value as an effective aspect as well as reducing time wastage, convenient, accurate and appropriate customer selection when purchasing. The main purpose of testing (simulating images) is to help customers check the size, fit and overall appearance of wearable products in the digital environment. In this paper, a model based on deep learning is presented for testing using a deep generator with a proposed method (PSO). In this method, after receiving the data set, it is divided into two parts, training and testing. The training part is provided by a Generative Adversarial Network (GAN). Then this trained network enters the optimization algorithm (PSO) to improve the weight of the neurons. The results showed that the proposed approach tried to improve the GAN neural network by relying on the meta-heuristic algorithm of particle swarm. Meta-heuristic algorithms have little complexity and have shown good performance in finding optimal points. Also, the proposed approach can significantly reduce costs and time and provide a better match between clothes and body shape of users. This system can be used as an effective tool in designing personalized and industrial clothing.

1. Introduction

Simulation of wearable products using 3D body scan data, accurate simulation of clothing fit based on physical characteristics of fabric and body is done. This technique seeks to transfer the clothes in one image to the desired person in another image. image, resulting in a realistic and acceptable composite image. The key to this is that, assuming that the synthetic results are realistic enough, the details of the clothing texture and other character traits of the subject (such as appearance and pose) should be well preserved. Recent advances in deep generative models, generative adversarial networks (GANs) [1] and their variants open a new door to countless mods Applications in clothing testing in the digital environment include fashion design, speech-assisted clothing generation [2, 3, 4], clothing appearance modification, and the like [5, 6]. Fashion design is very complex because it requires combining several pieces of clothing to create a harmonious ensemble. The type of garment texture or type of material may have very different appearance such as texture and color (for example, cotton shirt, jeans, leather shoes, etc.). But when put together they complement each other and form a stylish ensemble for a person. In order to create engaging virtual experiences and impressive fashion designs, it is important to consider the coordination and compatibility between garments as a whole rather than examining each garment separately. But modeling this compatibility in digital garment testing contexts is challenging because there is no accurate data to show whether the garments are compatible or not. Therefore, researchers use the coordination between clothing texture types, or synchronicity between clothing, as a weak cue to measure compatibility [7,8,9]. Much of the previous virtual experimental research was based on generative adversarial networks (GAN) to generate more realistic images. To preserve more details, previous studies [10-14] used a precision

deformation module that matched the target garment to the human body. After adjusting the morphed clothing, it was fed into a generator to create an image of the person independent of the clothing to produce the final result. However, the efficiency of this framework is highly dependent on the quality of the configured garments. Poor quality clothing can prevent accurate images from being produced. In Although these works have yielded positive results, they exist. To improve the GAN performance of the proposed PSO algorithm which is a biologically inspired algorithm influenced by the social behavior of the herd or fish training. It should be emphasized that, unlike other genetic algorithms, the PSO algorithm does not use an evolutionary operator such as crossover or mutation, and there are only a few parameters to change, which makes it easily applicable and also provides the best global solution for the optimization problem at hand. Gives [15].
Particle Swarm Optimization (PSO) is an innovative intelligent optimization algorithm that is easy to implement and has been widely used in the field of optimization to solve nonlinear systems and can be real and complex roots of a system. In order to improve the research accuracy and reduce the output error of GAN neural network, in this research, the proposed algorithm which is a combination of PSO-GAN has been used to detect the appropriate weights and biases in the network.

2. Related Work

The first popular image-based virtual try-on model builds upon a coarse-to fine network. First, it predicts a coarse image of the reference person wearing the try-on garment, then it refines the texture and shape of the previously obtained result. Wang et al. [16] overcame the lack of shape-context precision (i.e. bad alignment between clothes and body shape) and proposed a geometric transformation module to learn the parameters of a thin-plate spline transformation to warp the input garment. Following this work, many different solutions were proposed to enhance the geometric transformation of the try-on garment. For instance, Liu et al. [17] integrated a multi-scale patch adversarial loss to increase the realism in the warping phase. Minar et al. and Yang et al. [18] proposed different regularization techniques to stabilize the warping process during training.Instead, other works focused on the design of additional projections of the input garment to preserve details and textures of input clothing items. Another line of work focuses on the improvement of the generation phase of final try-on images. Among them, Issenuth et al. [19] introduced a teacher-student approach: the teacher learns to generate the try-on results using image pairs (sampled from a paired dataset) and then teaches the student how to deal with unpaired data. This paradigm was further improved in with a student-tutor-teacher architecture where the network is trained in a parserfree way, exploiting both the tutor guidance and the teacher supervision. On a different line, Ge et al. presented a self-supervised trainable network to reframe the virtual try-on task as clothes warping, skin synthesis, and image composition using a cycle-consistent framework.[20]

YAO FENG, 2019, addressed this issue in an article entitled Capturing and Animation of Body and Clothing from Monocular Video. In this research, they noted that hybrid modeling enables headscarves to (i) animate wearing avatars by changing body postures (including hand expressions and facial expressions), (ii) combining novel avatar views, and (iii) Transferring clothes in virtual avatars. Our experimental programs show that SCARF clothes with a higher visual quality than existing methods, with which the clothes change the shape of the body and body shape, and the clothes can be It will change. Achievements are transferred between avatars of different subjects.[20]

Jianbin Jiang, 2021, addressed this issue in a research entitled ClothFormer: Taming Video Virtual Try-on in All Module. In particular, ClothFormer involves three major modules. First, a two-stage anti-occlusion warping module that predicts an accurate dense flow mapping between the body regions and the clothing regions. Second, an appearance-flow tracking module utilizes ridge regression and optical flow correction to smooth the dense flow sequence and generate a temporally smooth warped clothing sequence. Third, a dual-stream transformer extracts and fuses clothing textures, person features, and environment information to generate realistic try-on videos. Through rigorous experiments, we demonstrate that our method highly surpasses the baselines in terms of synthesized video quality both qualitatively and quantitatively[21].Benjamin Fele et al., 2022, addressed this issue in a paper entitled C-VTON: Context-Driven Image-Based Virtual Try-On Network. Experimental results show that the proposed approach is able to produce photo-realistic and visually convincing results and significantly improves on the existing state-of-the-art.Sen He, et al.,2019, addressed this issue in a research.In this paper, we have proposed a style based global appearance flow estimation method to

warp the garment for virtual try-on. Our method via style modulation first estimates the appearance flow globally and then refines the appearance flow locally. Our method achieves state-of-theart performance on the VITON benchmark and it is more robust against large misalignment between person and garment images, as well as difficult poses/occlusions. We conducted extensive experiments to show the superiority of our method and validated our architecture design [22]. Xinyue Zhou, 2018, in research entitled Cross Attention Based Style Distribution for Controllable Person Image Synthesis addresses this issue. This paper presents a cross attention-based style distribution block for a singlestage controllable person image synthesis task, which has strong ability to align the source semantic styles with the target poses. The cross attention-based style distribution block mainly consists of self and cross attention, which not only captures the source semantic styles accurately, but also aligns them to the target pose precisely. To achieve a clearer objective, the AMCE loss is proposed to constrain the attention matrix in cross attention by target parsing map. Extensive experiments and ablation studies show the satisfactory performance of our model, and the effectiveness of its components. Finally, we show that our model can be easily applied to virtual try-on and head(identity) swapping tasks.[23]

3. Methodology

Over the past decade, interest in the virtual "Try-On" has grown and is recognized for its value as an effective aspect of the customer experience. Its main purpose has been to help customers check the size and fit of wearable products virtually, and it has become an enjoyable experience for the customer. Because the virtual try-on can help customers filter out incorrect sizes and fits, this experience has become very useful in online shopping malls. In this thesis, we have presented a model based on deep learning for virtual try-on using deep generator.

Figure (1) shows the flowchart of the proposed method. As it is clear in this figure, the proposed approach first receives the data set and then divides this data set into two parts, training and testing. The training part is provided by a GAN deep neural network system for training. After that, this trained neural network enters the Particle Swarm Optimization (PSO) algorithm in order to improve the weight of neurons. In final step we evaluate the proposed method using test part of data.

As we know the standard method for training neural networks is the method of stochastic gradient descent (SGD). The problem of gradient descent is that in order to determine a new approximation of the weight vector, it is necessary to calculate the gradient from each sample element, which can greatly slow down the algorithm and decrease the accuracy of neural network. Therefore, in the proposed approach, we improve the weight of neurons using the particle swarm algorithm. Next, we examine the structure of Generative neural networks (GAN).

3. 1. GAN

GAN are a type of deep neural network used to generate artificial images. This architecture consists of two deep neural networks, a generator and a discriminator, which work against each other. The generator generates new data samples, while the discriminator evaluates the data for correctness and decides whether each data sample is "real" from the training dataset or "fake" from the generator.The generator and discriminator are trained together to work against each other until the generator is able to create real synthetic data that the discriminator can no longer detect as fake. After successful training, the data generated by the generator can be used to create new synthetic data, for potential use as input to other deep neural networks. GANs are versatile because they can learn new examples from any type of data, such as synthetic images of faces, new songs in a certain style, or lyrics of a certain genre. Therefore, they are very suitable for creating images in online stores. These networks can combine the images of customers and the desired product with each other and the output of this system will be clothing by the customer. This neural network consists of two generator and discriminator networks. These two sections are described below:

1. Generator: Given a vector of random values (latent inputs) as input, this network generates data with the same structure as the training data.
2. Discriminator: Given batches of data containing observations from both the training data, and generated data from the generator, this network attempts to classify the observations as "real" or "generated"[24].

3.2. Improve GAN hyper-parameters with PSO

PSO is a bio-inspired algorithm influenced by flock social behavior or fish training. It should be emphasized that, unlike other genetic algorithms, the PSO algorithm does not use an evolutionary operator such as crossover or mutation, and there

are only a few parameters to alter, making it easily applicable. In the PSO algorithm, birds or fish that represent particles fly through the issue space to identify the optimal quantity of cost or target performance by learning the current optimal particles. In this case, the target cost or performance symbolizes the food that should be located in the search / issue space for the birds or fish. PSO begins with a large number of random particles (solutions) and then iteratively seeks for Optima by updating generations. In PSO, each particle can preserve the best position known as the best universe or Gbest discovered by the entire swarm in history, as well as the best position known as the best person or Pbest discovered by each particle. Learning particles from the Gbest and Pbest positions yields the best global solution to the optimization issue.

Best Personal Memory (PBest): As it approaches towards the answer, each particle recalls the best place it has visited as the best personal memory. In particle motion, this memory may be updated.

Best Collective Memory (GBest): The particle swarm algorithm keeps the best memory available to all particles. Once the entire particle has migrated, this memory may be updated.Current position vector: Each particle is identified by a vector that specifies the particle's current location.Speed vector: In addition to the current position vector, each particle has a velocity vector that describes the direction and speed of the following particle's motion.Figures (2) depict the particle location vector, velocity vector, and current position vector following motion. The position of a particle changes when the cell corresponding to that position is in the velocity vector 1, as seen in this illustration. In this scenario, if the desired cell's quantity is zero, it will be changed to one, and if it is one, it will be changed to zero.

Particle position vector	Position of the first particle
X1	1
X2	1
X3	0
X4	1
X5	0
X6	1
X7	0
X8	1

Particle velocity vector	First particle speed
	0
	0
	1
	1
	0
	0
	0
	0

New particle position vector	New position of the first particle
	1
	1
	1
	0
	0
	1
	0
	1

Figure 2. *Current position vector, velocity vector and particle position vector after motion*

In PSO, every particle or individual in the population is a potential solution. Particles are grouped in a multidimensional search space, where the position of each particle is adjusted according to its own and its neighbors experience. All particles have proportional values that are calculated by the target function and optimized by the PSO algorithm and have accelerations that guide the motion of the particles. For example, $x_i\ (t)$ indicates the position of the *i* particle in the search space at time t. The position of the particle changes to the state $x_i\ (t + 1)$ by adding the acceleration $v_i\ (t + 1)$ to the current position, i.e.

$$x_i(t + 1) = x_i(t) + v_i(t + 1) \quad (1)$$

The starting population is generated in such a way that the particles are spread randomly across the search space. The acceleration vector directs the optimization process and represents the empirical knowledge of the particles as well as the social information transferred between the particles' neighbors. The empirical knowledge of a particle is generally referred to as the cognitive component, which is proportional to the distance of the particles from their best position called pbest, which has been found since the first step. The social information exchanged is referred to as the social component of the velocity equation. The social component of particle acceleration updates the reflected information obtained from all particles in the congestion.In this case, social information is the best place to get through crowds and is called gbest. In each repetition, each particle is updated according to these two values of pbest and gbest. For gbest PSO, the particle acceleration i is calculated as follows:

$$V_{ij}(t + 1) = iw \times V_{ij}(t) + c_1 r_{1j}(t)\left(y_{ij}(t) - X_{ij}(t)\right) + c_2 r_{2j}(t)\left(\hat{y}_j(t) - X_{ij}(t)\right) \quad (2)$$

Where iw is the weight factor, $V_{ij}(t)$ is the velocity of the particle i in dimensions j = 1,…, nx at time t. $X_{ij}(t)$ is the position of the particle i in dimensions

j in the time step t, c_1 and c_2 are positive acceleration constants used for the scale of cognitive and social segments, respectively, $r_{1j}(t)$, $r_{2j}(t)$ ~ U (0,1) Random values in the range [0, 1] that are sampled from a uniform distribution. The best personal position y_{ij}, which is associated with the i particle, is the best particle location in the first step, called lbest or local best. According to the minimization problems, the best personal position in the next time t + 1 is calculated as follows:

$$y_i(t+1) = \begin{cases} y_j(t) & if\ f(X_i(t+1)) \geq f(y_j(t)) \\ X_i(t+1) & if\ f(X_i(t+1)) < f(y_j(t)) \end{cases} \quad (3)$$

Where f can be a target function in PSO or several target functions in MPSO that measures whether the solution is optimal, that is, it measures the performance or quality of a particle (or solution). The best global position (gbest) in the time step t, y '(t), is defined as follows:

$$\hat{y}(t)_c - \{y_0(t).\cdots\cdots.y_{ns}(t)\}|f(\hat{y}(t)) = \min\{f(y_0(t)).\cdots\cdots.f(y_{ns}(t))\} \quad (4)$$

Where ns is the total number of particles in the swarm. The PSO process is repetitive. After generating an initial congestion, the value of the proportionality function in each repetition is evaluated and the acceleration and position of each particle are updated accordingly. The algorithm terminates when one of the following happens:

1. The number of repetitions has reached its maximum.
2. An acceptable solution has been found.
3. No improvement was observed in a number of repetitions.

Particle swarm algorithm is a very efficient algorithm and can remember the best global value and affect the motion of other particles in such a way that it causes faster convergence and falls into local optimal solutions. It should be noted that in this particle swarm algorithm seek to optimize only one goal.

In general, the PSO algorithm consists of a group of components that move in a multidimensional search space with real values of possible problem solutions. PSO is easy to implement and computationally is low cost. PSO is also effective in solving many GO problems and in some cases does not run into problems with other evolutionary computing techniques. One of the disadvantages of this method is the difficulty of setting the PSO parameters to achieve good performance, and if the parameters are not selected properly, the PSO tends to the local optimality and suffers early convergence.

3.3. Rule of PSO

Since the efficiency of the neural network algorithm and deep learning to select the initial values of the network weight, the biases and parameters in the algorithm depend on the learning rate. Therefore, to make these algorithms more robust, it is necessary to have an optimization algorithm that can select the appropriate initial weight and bias values for better network execution based on several parameters and being in the competency function. Thus, in many studies, researchers have used meta-heuristic algorithms instead of traditional algorithms to train neural networks. Meta-heuristic algorithms are more efficient than traditional algorithms and have more accuracy. In the field of virtual "Try-On" and to reduce the error rate by GAN neural networks, the selection of weight and bias parameters can be done using optimization algorithms to increase accuracy. The Particle Swarm Optimization (PSO) is an innovative intelligent optimization algorithm that is easy to implement and has been widely used in the field of optimization to solve nonlinear systems and can be the real and complex roots of a system. In order to improve the accuracy of the research and reduce the output error of the GAN neural network, in this research, the proposed algorithm, which is a combination of PSO -GAN, is used to detect the appropriate weights and biases in the network. The proposed hybrid algorithm has not been used in this field so far. The introduced method can use the following process:

- ✓ The weight and bias of artificial neural networks are encoded as a vector. This vector is a particle (a member of the particle swarm optimization algorithm population).
- ✓ Each particle (i.e. weight and bias vector) is analyzed with the cost function or the average detection error rate per replicate. The PSO optimization algorithm updates the vectors in each iteration.
- ✓ In the last iteration, the optimal weight and bias vector are used to learn how to use a GAN neural network to optimize the output rate of the model.
- ✓ The initial population starts from random solutions in the PSO population, which considers a GAN neural network as a particle.

- ✓ Society is analyzed by mean error.
- ✓ The PSO model is applied to neural networks and weights and biases are updated.
- ✓ The optimal GAN neural network is selected in each iteration.
- ✓ The best GAN network is extracted with optimal weight and bias.

3. 4. SSIM

Structural similarity index measurement (SSIM) is a method for predicting the perceived quality of a variety of digital images and videos. It is also used to measure the similarity between two images. The SSIM index is a perfect reference metric. In other words, the measurement or prediction of image quality is based on an uncompressed or undistorted original image as a reference. SSIM is a perception-based model that considers image degradation as perceived change in structural information, while also incorporating important perceptual phenomena, including both luminance masking and contrast masking terms. The difference with other techniques such as MSE or PSNR is that these approaches estimate absolute errors. Structural information is the idea that the pixels have strong inter-dependencies especially when they are spatially close. These dependencies carry important information about the structure of the objects in the visual scene. Luminance masking is a phenomenon whereby image distortions (in this context) tend to be less visible in bright regions, while contrast masking is a phenomenon whereby distortions become less visible where there is significant activity or "texture" in the image.

$$l(x,y) = \frac{2\mu_x\mu_y + c_1}{\mu_x^2 + \mu_y^2 + c_1}$$

$$c(x,y) = \frac{2\sigma_x\sigma_y + c_2}{\sigma_x^2 + \sigma_y^2 + c_2} \quad (5)$$

$$(6)$$

$$s(x,y) = \frac{\sigma_{xy} + c_3}{\sigma_x\sigma_y + c_3} \quad (7)$$

3.5. FID

First Imput Delay (FID) is an important and user-oriented feature, actually the duration of the first user voting with the site. Bad experience that users can experience when trying to create work with non-responsive pages. When the FID value is low, you can be sure that your page is usable.

4. Experiments

4.1. Experiment Setup

VITON 5 dataset:

We test our model on the VITON 5 dataset [24]. This dataset is among the most popular datasets used in previous virtual try-on (VTON) works. VITON contains a set of 16235 pairs of images. Both person and clothing images are 256 x 192 resolution. Each pair means the image of the person and the image of the clothes on the person. The figure below shows a part of this data set with paired images.

4.2Evaluation criteria

In addition to implementing the proposed method, as well as a compared method and having a benchmark data set, evaluation criteria are needed to conduct tests. In fact, evaluations make sense in the form of these evaluation criteria, and the comparison of different methods should be based on these criteria. In this research, we use the famous Structure Similarity Index Measure (SSIM) to evaluate the proposed method. The SSIM index is calculated on different windows of an image. The size between two windows x and y is considered to be $N\times N$. The SSIM index is based on the calculation of three terms, i.e. brightness, contrast and structure of two images. The SSIM index is a multiplicative combination of these three terms. Below are the equations related to this criterion.

$SSIM(x,y)=[l(x,y)]^{\alpha}\cdot[c(x,y)]^{\beta}\cdot[s(x,y)]^{\gamma}$

Where

$$l(x,y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}$$

$$l(x,y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}$$

$$l(x,y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3}$$

where μ_x, μ_y, σ_x, σ_y, and σ_{xy} are the local means, standard deviations, and cross-covariance for images *x*, *y*. If $\alpha = \beta = \gamma = 1$, and $C_3 = C_2/2$ (default selection of C_3) the SSIM simplifies to:

$$SSIM = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$$

In addition to our SSIM, Fréchet Inception Distance (FID) is also used to evaluate the proposed method. FID is a metric used to assess the quality

Figure 3. *some examples of input/output images of the proposed method*

of images created by a generative model, like a generative adversarial network (GAN). Unlike the earlier Inception Score (IS), which evaluates only the distribution of generated images, the FID compares the distribution of generated images with the distribution of a set of real images ("ground truth") [25]. The FID metric does not completely replace the IS metric. Classifiers that achieve the best (lowest) FID score tend to have greater sample variety while classifiers achieving the best (highest) IS score tend to have better quality within individual images [26]. The FID metric was introduced in 2017[25] and is the current standard metric for assessing the quality of generative models. It has been used to measure the quality of many recent models including the high-resolution.For any two probability distributions μ, ν over R^n which having finite mean and variances. Between the Gaussian with mean (m_μ, C_μ) obtained from μ and the Gaussian with mean (m_ν, C_ν) obtained from ν, the FID calculates using:

$$d^2((m_\mu, C_\mu), (m_\upsilon, m_\upsilon) = \left|\left|m_\mu - m_\upsilon\right|\right|_2^2 + Tr(C_\mu + C_\upsilon - 2(C_\upsilon C_\mu)^2$$

5. Results

As stated before, we will use two famous criteria, FID and SSIM, for evaluation. Therefore, in this section, we will present the evaluation results based on these criteria. Considering that the evaluation parameters can have a significant impact on the performance of each of the compared methods, we have considered these parameters. One of the most important parameters considered is how to divide the data between training and testing sets. In fact, in these evaluations, we performed evaluations for different sizes of training and testing datasets. The evaluations in this section are considered variable for the change in the size of the training data set between 40% of the data and 80%. To do this, for each change in the size of the training data set, the proposed method as well as the basic method are evaluated and its results are expressed in the form of four introduced criteria.The second effective parameter in conducting evaluations is the size of the data set. In fact, it is not always possible to expect that the entire data set will be available, so how will any of the compared methods be affected by changing the size of the data set? To answer this question, we have used the scenario of changing the size of the data set. In this scenario, we have changed the size of the data set between 50% and 100% and performed the tests. In this scenario, the size of the training data set is 70% and the size of the test data set is 30%. In the following, we will present the results of the two proposed scenarios in the form of the two evaluation criteria. It should be noted that after presenting the results in the form of both scenarios, we will analyze the results in the next section. The first criterion evaluated includes SSIM. The results of this criterion for both scenarios can be seen in the figures below. As it is clear in the results of these figures, the proposed method has been able to provide better results than

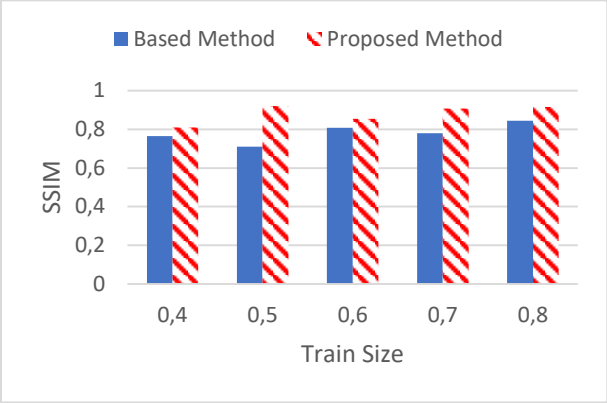

Figure 4. *Comparison of SSIM for the compared methods in the first scenario(Change the size of the dataset between 40% of the data and 80%)*

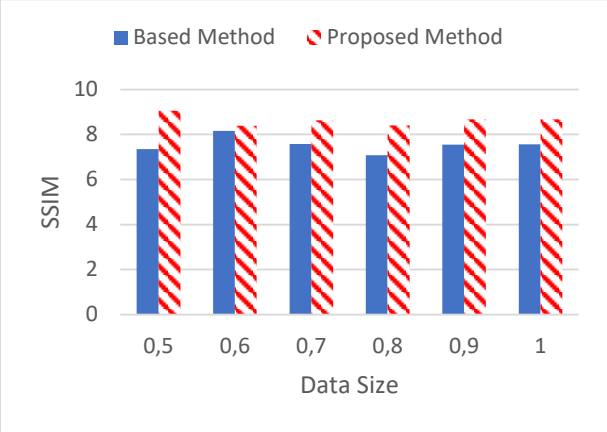

Figure 5. *Comparison of SSIM for the compared methods in the second scenario (Change the size of the dataset between 50% and 100%*

the compared method for all the measured intervals. In such a way that the change in the size of the training data set and the size of the data set could not affect the superiority of the proposed method compared to the compared method .The second criterion evaluated includes FID. The results of this criterion in both scenarios can be seen in figures (3-4) and (4-4).

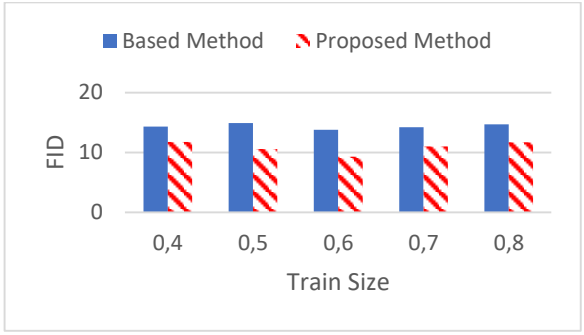

Figure 6. *Comparison of FID for the compared methods in the first scenario*

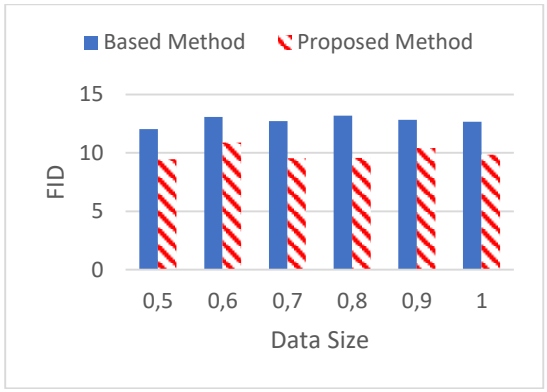

Figure 7. *Comparison of FID for the compared methods in the second scenario*

As it is clear from the results of this figure, the proposed method has always been able to show better results than the compared method. This issue is true in all the measured range and both scenarios. Examining the results presented in this Article shows that the proposed method has been able to obtain better results than the basic method for the change in the size of the training data set and the size of the data set i.e. scenarios 1 and 2 and for all the examined criteria. The superiority of the proposed method is calculated in the table (1). By comparing the proposed method with the basic method (Junhong Gou et al.'s study, Taming the Power of Diffusion Models for High-Quality Virtual Try-On with Appearance Flow), it can be

Table 1. *Comparison of Scenarios 1 and 2*

Improvement (%)	Evaluation criteria	Scenario
6 to 30	SSIM	**the first scenario(Change the size of the dataset between 40% of the data and 80%)**
18 to 33	FID	
3 to 23	SSIM	**in the second scenario(Change the size of the dataset between 50% and 100%)**
17 to 28	FID	

seen that the proposed approach has tried to improve the GAN neural network by relying on the particle swarm meta-heuristic algorithm. It should be noted that neural networks determine the weight of neurons and learning processes based on gradient descent approaches. Gradient descent is an optimization algorithm that adjusts synaptic weights in neural networks. In other words, with

the help of gradient descent, trained neural networks acquire the necessary knowledge to solve the problem. Neural networks like GAN use batch approaches to train the neural network. Despite its accuracy and consistency, batch gradient descent has several drawbacks. It can be slow and computationally expensive when dealing with large datasets and complex models. Additionally, it can get stuck in shallow local minima or saddle points, where the gradient is either very small or zero. These problems can challenge the neural network learning process and thus reduce its quality. Therefore, in the proposed method, in addition to using the batch gradient descent approach, we have also used meta-heuristic algorithm of particle swarm. Meta-heuristic algorithms have low complexity and have shown good performance in finding optimal points. The result of this approach can be seen in the results presented in this report.In the study by Khoei et al titled DM-VTON: Distilled Mobile Real-time Virtual Try-On is based on a knowledge distillation scheme that leverages a strong Teacher network as supervision to guide a Student network without relying on human parsing. Experimental results show that the proposed method can achieve 40 frames per second on a single Nvidia Tesla T4 GPU and only take up 37 MB of memory while producing almost the same output quality as other state-of-the-art methods. DMVTON stands poised to facilitate the advancement of real-time AR applications, in addition to the generation of lifelike attired human figures tailored for diverse specialized training tasks [30]. In comparing this article with Khoyi's research, both papers focus on improving virtual clothing testing systems, but they employ different methods and offer distinct applications. Khoyi's research primarily emphasizes real-time applications, delivering a lightweight and fast solution with a speed of 40 frames per second and optimized memory usage. If accuracy and personalization are the main goals, our research model, due to the use of PSO for optimizing GANs, could be a better option. This method is suitable for precise and industrial designs requiring high quality. On the other hand, if real-time speed and efficiency are the priority, the DM-VTON paper excels. Its ability to run at high speed with low memory consumption makes it ideal for interactive and AR-based applications. Additionally, it can be stated that while our proposed approach is more suitable for industrial and fashion design applications, DM-VTON is superior for interactive and real-time systems.Also, David's research entitled LaDI-VTON: Latent Diffusion Textual-Inversion Enhanced Virtual Try-On that This work introduces LaDIVTON, the first Latent Diffusion textual Inversion-enhanced model for the Virtual Try-ON task. The proposed architecture relies on a latent diffusion model extended with a novel additional autoencoder module that exploits learnable skip connections to enhance the generation process preserving the model's characteristics. To effectively maintain the texture and details of the in-shop garment, we propose a textual inversion component that can map the visual features of the garment to the CLIP token embedding space and thus generate a set of pseudo-word token embeddings capable of conditioning the generation process. Experimental results on Dress Code and VITON-HD datasets demonstrate that our approach outperforms the competitors by a consistent margin, achieving a significant milestone for the task [31].In comparing our study with David's research: The focus and main objective of David's study, as presented in the LaDI-VTON paper, is to improve the virtual try-on system using Latent Diffusion Models (LDM) and combining Textual Inversion techniques. The primary goal of this research is to preserve the details and texture of clothing items available in stores while offering an advanced model with precise conditioning capabilities for image generation. In contrast, our research focuses on using the Particle Swarm Optimization (PSO) algorithm to enhance the performance of deep generative adversarial networks (GANs). The aim is to develop a system that achieves higher accuracy in matching clothing to users' body shapes, while reducing the cost and time involved in designing personalized clothing. If the goal is to preserve visual details and generate highly accurate images aligned with store data, David's research would be a better choice. If the objective is personalized clothing design, cost and time reduction, and achieving a precise match between clothing and users' body shapes, our research offers greater advantages. It can also be said that both studies address different goals and applications.

David's research excels in systems based on pre-existing data (such as online stores), whereas our research is more suited for personalized and industrial clothing design. The choice of the best method depends on the specific needs and intended application.

Author Statements:

- **Ethical approval:** The conducted research is not related to either human or animal use.
- **Conflict of interest:** The authors declare that they have no known competing financial interests or personal relationships that could

have appeared to influence the work reported in this paper

- **Acknowledgement:** The authors declare that they have nobody or no-company to acknowledge.
- **Author contributions:** The authors declare that they have equal right on this paper.
- **Funding information:** The authors declare that there is no funding to be acknowledged.
- **Data availability statement:** The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

References

[1] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. In *NeurIPS*.

[2] Zhang, Y., Jauvin, C., & Pal, C. (2018). Fashion-gen: The generative fashion dataset and challenge. *arXiv preprint arXiv:1806.08317*.

[3] Zhu, S., Fidler, S., Urtasun, R., Lin, D., & Chen, C. L. (2017). Be your own prada: Fashion synthesis with structural coherence. In *ICCV*.

[4] Zanfir, M., Popa, A. I., Zanfir, A., & Sminchisescu, C. (2018). *Human appearance transfer*.

[5] Gunel, M., Erdem, E., & Erdem, A. (2018). *Language guided fashion image manipulation with feature-wise transformations*.

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

[7] Han, X., Wu, Z., Jiang, Y. G., & Davis, L. S. (2017). Learning fashion compatibility with bidirectional lstms. In *ACM Multimedia*.

[8] Veit, A., Kovacs, B., Bell, S., McAuley, J., Bala, K., & Belongie, S. (2015). Learning visual clothing style with heterogeneous dyadic co-occurrences. In *CVPR*.

[9] Hsiao, W. L., Katsman, I., Wu, C. Y., Parikh, D., & Grauman, K. (2019). Fashion++: Minimal edits for outfit improvement. In *ICCV*.

[10] Ge, Y., Song, Y., Zhang, R., Ge, C., Liu, W., & Luo, P. (2021). Parser-free virtual try-on via distilling appearance flows. In *CVPR*.

[11] Han, X., Hu, X., Huang, W., & Scott, M. R. (2019). Clothflow: A flow-based model for clothed person generation. In *ICCV*.

[12] He, S., Song, Y. Z., & Xiang, T. (2022). Style-based global appearance flow for virtual try-on. In *CVPR*.

[13] Minar, M. R., Tuan, T. T., Ahn, H., Rosin, P., & Lai, Y. K. (2020). Cp-vton+: Clothing shape and texture preserving image-based virtual try-on. In *CVPR Workshops*.

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

[15] Brock, A., Donahue, J., & Simonyan, K. (2018). Large scale GAN training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*.

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

[17] Liu, Z., Luo, P., Qiu, S., Wang, X., & Tang, X. (2016). DeepFashion: Powering robust clothes recognition and retrieval with rich annotations. In *CVPR*.

[18] Yang, H., Zhang, R., Guo, X., Liu, W., Zuo, W., & Luo, P. (2020). Towards photo-realistic virtual try-on by adaptively generating-preserving image content. In *CVPR*.

[19] Issenhuth, T., Mary, J., & Calauz`enes, C. (2020). Do not mask what you do not need to mask: a parser-free virtual try-on. In *ECCV*.

[20] Choi, S., Park, S., Lee, M., & Choo, J. (2021). VITON-HD: High-resolution virtual try-on via misalignment-aware normalization. In *ICCV*.

[21] Cui, A., McKee, D., & Lazebnik, S. (2021). Dressing in order: Recurrent person image generation for pose transfer, virtual try-on and outfit editing. In *ICCV*.

[22] Dong, H., Liang, X., Shen, X., Wang, B., Lai, H., Zhu, J., Hu, Z., & Yin, J. (2019). Towards multi-pose guided virtual try-on network. In *ICCV*.

[23] Fenocchi, E., Morelli, D., Cornia, M., Baraldi, L., Cesari, F., & Cucchiara, R. (2022). Dual-branch collaborative transformer for virtual try-on. In *CVPR Workshops*.

[24] Fincato, M., Landi, F., Cornia, M., Fabio, C., & Cucchiara, R. (2020). VITON-GT: An image-based virtual try-on model with geometric transformations. In *ICPR*.

[25] Bińkowski, M., Sutherland, D. J., Arbel, M., & Gretton, A. (2018). Demystifying MMD GANs. In *Proceedings of the International Conference on Learning Representations*.

[26] Dhariwal, P., & Nichol, A. (2021). Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems*.

[27] Han, X., Wu, Z., Yu-Gang, J., & Davis, L. S. (2018). Viton: An image-based virtual try-on network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[28] Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., & Hochreiter, S. (2017). GANs trained by a two time-scale update rule converge to a local Nash Equilibrium. In *Advances in Neural Information Processing Systems, 30*. arXiv:1706.08500.

[29] Ho, J., & Salimans, T. (2022). *Classifier-Free Diffusion Guidance*. arXiv:2207.12598.