Bulletin of Networking, Computing, Systems, and Software – www.bncss.org, ISSN 2186-5140
Volume 13, Number 1, pages 16–22, January 2024

Improving the Accuracy of Virtual Try-On with Real-World Dataset Using Genetic Algorithm

1st Kosei Takaki
National Institute of Technology,
Oita College, Japan
Electrical Electronics Information Engineering Major
Oita,Japan
aes2307@oita.kosen-ac.jp

2nd Takumi Ikenaga
Nara Institute of Science and Technology
Division of Information Science
Department of Science and Technology
Nara,Japan
ikenaga.takumi.ir8@is.naist.jp

3rd Taiyo Sato
National Institute of Technology,
Oshima College,Japan
Electronics Infomation Technology System
Yamaguchi,Japan
d23003@oshima.kosen-ac.jp

4th Shudai Ishikawa
National Institute of Technology,
Oita College, Japan
Department of Computer Science
Oita,Japan
shu-ishikawa@oita-ct.ac.jp

***Abstract*—Virtual try-on refers to trying on clothing virtually on a computer. The latest research has proposed highly accurate try-on methods that can complement any posture and features of clothing (such as texture and logo). However, try-on an input image that significantly differs from the training data is difficult. In other words, if images input by the user (images taken by a web camera or smartphone that show the background) are used as input data, they cannot be processed appropriately. This research aims to realize virtual try-on under natural environment conditions, using images taken by web cameras or smartphones as input. Creating a dataset of sufficient volume for training requires a lot of time and effort. To learn efficiently with little training data, we apply transfer learning optimized by a genetic algorithm (GA), which optimizes the layers of a pre-trained model and additional models. Using GA to determine the weight update and fixation of the network during training, we aim to realize learning with less data. Virtual try-on systems consist of several networks responsible for partitioning human images, deforming clothes, and generating final output images. If retraining is performed for all networks, the number of networks and data do not match, leading to overfitting. Using GA, we carefully select networks in advance and verify the accuracy of the virtual try-on. The try-on results were compared and evaluated using SMD in conventional GA methods with the teacher data. However, preparing teacher data is costly and time-consuming, as it uses images not included in the existing training dataset, similar to a natural environment dataset. Therefore, we developed a method that evaluates only the try-on results, compared it with the method using teacher data, and verified it.**

***Index Terms*—Deep Learning, Virtual Try-On, Human Segmentation,GA**

I. Introduction

In recent years, the E-commerce market in the fashion industry has been expanding rapidly. While purchasing clothing online can significantly reduce the effort and time required to visit physical stores, it poses challenges such as mismatched coordination or differences from the depicted photos due to the inability to try on the clothes. Therefore, virtual try-on method, which involves transferring clothing features (textures, logos) onto the user's clothing area using computer algorithms, has gained attention. Existing virtual try-on methods have mainly focused on addressing complex body poses and maintaining clothing features, with little mention of practical applications [1]–[4]. ACGPN (Attribute-Controlled Generative Pose Network) is a virtual try-on method that improves try-on accuracy by dividing the process into the target person's preservation area, clothing area, and generated area [3]. ACGPN comprises several networks responsible for clothing deformation, target person segmentation, and image generation. Models trained on pre-existing datasets (such as the VTON dataset) in ACGPN demonstrate high accuracy in ideal conditions for try-on. However, achieving proper try-on for input images that differ significantly from the training data, such as those captured by web cameras or smartphones, is challenging. Therefore, attempts were made to retrain the model using real-world datasets created from images captured by web cameras or smartphones. However, due to the limited training data, issues such as failure to preserve clothing features or overfitting remained. Creating a dataset with a sufficient data points for training requires significant time and effort. Transfer learning is a method that enables efficient learning with limited data. Transfer learning performance in neural networks has been reported to depend on selecting network layers for retraining [5], [6]. While layer selection is typically performed manually, networks performing complex tasks like virtual try-on involve many layers, making the selection of layers for retraining difficult. In ACGPN, transfer learning is achieved by loading pre-trained models, setting the network weights, and further layer selection becomes even more challenging due to the composition of multiple networks.

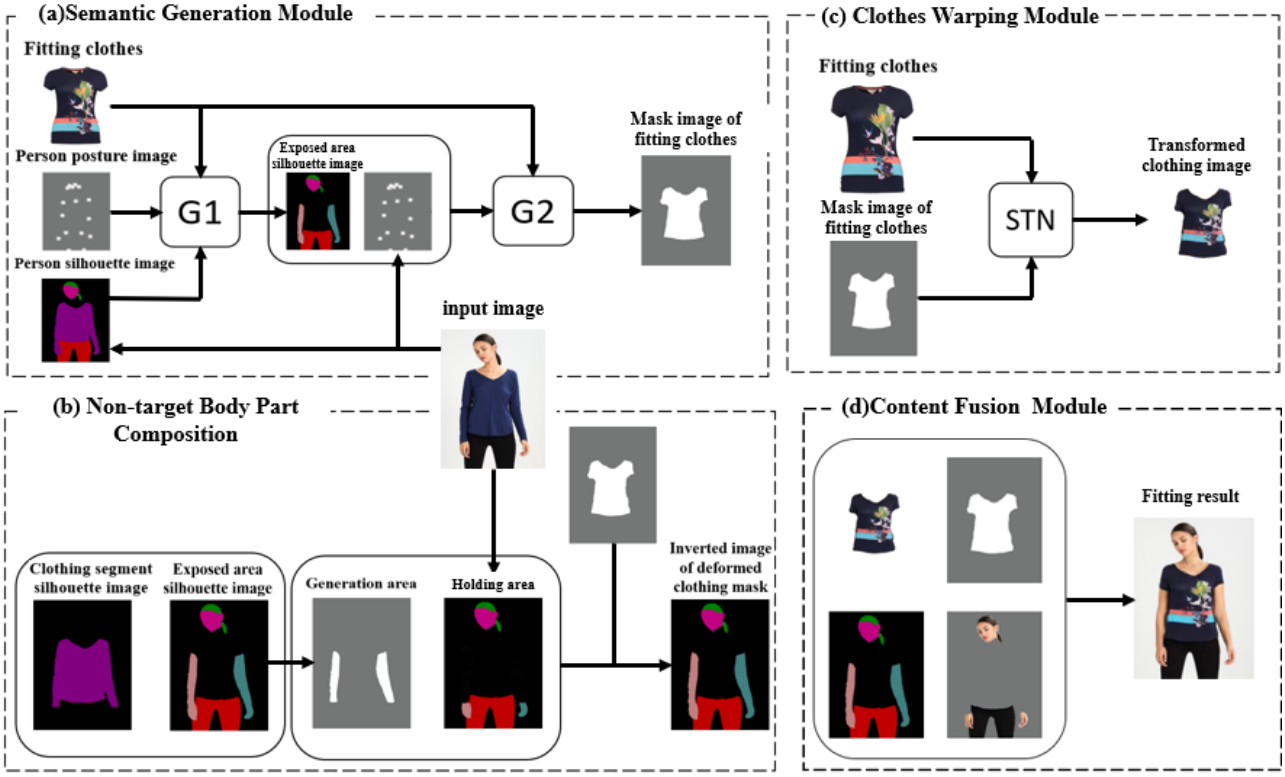

Fig. 1. Virtual try-on system flow

Specifically,we apply transfer learning by optimizing the pre-trained and additional model's layer using GA [7].By controlling the weight updates and fixations of the network layers during training, we aim to achieve efficient learning with a limited amount of data. In the experiments, we perform transfer learning from the VTON dataset to the pre-trained model using the real-world dataset and specify the networks for weight updates and fixations using arrays. We generate arrays with many elements corresponding to the number of networks and search for the best individual in the GA population. The evaluation of individuals is quantified by measuring the distances between the generated images and the target try-on images, and the distances between the generated clothing areas and the try-on clothing images. We validate the effectiveness of the evaluation method by comparing the generated images with the training data, comparing the generated clothing areas with the try-on areas, and comparing the re-dressed generated images. We also discuss the results of the try-on and provide insights.

II. Virtual try-on

A. ACGPN

Virtual try-on refers to virtually trying on clothes using computer algorithms. Various methods have been developed, such as VITON and PF-AFN [1], [4]. VITON deforms the clothing model to fit the shape of the target person, generating rough try-on images, reconstructing clothing features, and generating detailed try-on images. However, reconstructing the clothing model, it cannot fully restore the clothing model's features, resulting in blurry try-on results. PF-AFN is characterized by not requiring a segmentation mechanism for target person region delineation. However, when not performing region delineation, it requires training data from various environments. Considering practicality, training that encompasses all possible environments is not realistic. In this study, we adopted ACGPN, proposed by Han et al. ACGPN performs target person segmentation as a pre-processing step [3]. It is a method that improves try-on accuracy by dividing the clothing model deformation and dressing process into the preservation area of the target person,the try-on clothing area, and the generated area.It consists of four modules: determination of the try-on clothing area, deformation of the try-on clothing, extraction of the preservation area of the target person, and assembly of generated parts. The images are generated using a Generative Adversarial Network (GAN). Fig.1 illustrates the architecture of ACGPN.

Preprocessing:

The input data is obtained by preprocessing the raw data using SCHP [8] and Openpose [9]. This preprocessing generates try-on person images, person parsing images, person pose information, and try-on clothing data.

Semantic Generation Module(SGM):

Segmentation is performed on the pre-try-on image and try-on clothing image to determine the try-on clothing area. In

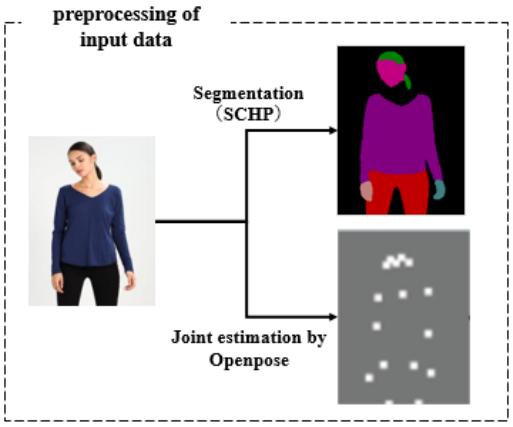

Fig. 2. Preprocessing of Input Data

Fig.1(a), G1 and G2 represent GAN (Generative Adversarial Network). A GAN consists of a Generator and a Discriminator. The Generator learns to generate the desired image given random input data. G1 takes try-on clothing, person pose information, and fused parsing image as input and generates a parsing image representing the exposed parts of the body. G2 takes try-on clothing, person pose information, and the parsing image representing the exposed parts generated by G1 as input and generates a mask image for deforming the clothing.

Clothes Warping Module(CWM):

In Fig.1(b), the try-on clothing is deformed to match the try-on clothing area received from SGM. Thin Plate Spline (TPS) transformation is used for deformation. TPS transformation treats the image as a thin plate and calculates how the surrounding pixels move when a certain pixel is moved to another coordinate.

Non-target Body Part Composition:

In Fig.1(c), based on the pre-try-on image and the person segmentation generated by SGM, the deformed try-on target person is divided into preservation area and generated area. Parsing images for the preservation area are generated from the clothing area parsing image and the exposed parts parsing image. The inverted mask image for deformed clothing is generated, combined with the try-on clothing mask image.

Content Fusion Module(CFM):

In Fig.1(d), the generated area is generated, and the preservation area, try-on clothing area, and generated area are assembled. The deformed clothing, preservation area, and generated area generated by CWM (Clothing Warping Module) and Non-target Body Part Composition are input to the GAN network, and the final try-on result is output.

III. Dataset

A. Clothing extraction mechanism

The clothing extraction is performed by utilizing the fact that the Upper-clothes, which is the clothing model target, is output as blue (0, 0, 128). The flow of the clothing extraction mechanism is shown in Fig.2. When the clothing model target is input to the clothing extraction mechanism, the person parsing results, obtained through SCHP, provide segmentation information.Keeping only the blue(0,0,128) parts from the person parsing results creates a clothing extraction mask. By applying the mask to the clothing model target, only the clothing can be extracted from the person's image. The extracted clothing is then filled with white around its edges and resized to fit the format of a virtual try-on, thus creating the clothing model.

B. Dataset

Models trained on pre-existing datasets like the VTON [1] enable high-accuracy try-on in ideal conditions. However, they have difficulty fitting images input by users, such as those taken with smartphones. In other words, fitting accuracy in real-world environments is not guaranteed. The authors of [10] created a real-world dataset for the practical application of virtual fitting. This dataset was compiled using images of 10 men, 40 short-sleeved clothing items, and 8 types of cameras, including webcams and smartphones. The subject images are those of the photographed individuals with the background removed, and the clothing models are further processed to extract only the clothing. The dataset consists of paired data of the subject and the clothing model. Below are the features of the two datasets.

VTON Dateset

The dataset used in this study was obtained from the fashion E-commerce site Zalando (https://www.zalando.de) and consists of 16,000 images of people and clothing.

Real-world dataset

The dataset was captured using web cameras and smartphones, and it consists of 1,900 images of people and clothing.

IV. Evaluation of generated images

A. Evaluation of generated images

In GA, the evaluation value is calculated based on the try-on images generated by the trained model. The evaluation of try-on images is performed using the Sliced Wasserstein Distance (SWD), a metric for image similarity evaluation [11]. Other evaluation metrics for generated images using GANs include Inception Score and Frechet Inception Distance (FID). Inception Score and FID use the Inception-V3 image recognition model for evaluation. However, since Inception-V3 is trained on the ImageNet dataset, it may not be proficient in feature extraction for clothing-related images. On the other hand, SWD is independent of the Inception model, making it a suitable choice for evaluation, as it ensures consistent evaluation regardless of the image content. Three evaluations are performed on the generated images: comparison with the training data, comparison of the clothing area with the try-on area, and comparison of the re-dressed images.

B. Re-dressing

Re-dressing refers to dressing the target person in the same clothing model they are wearing, enabling error analysis before and after try-on. This process allows for evaluation as an indicator.

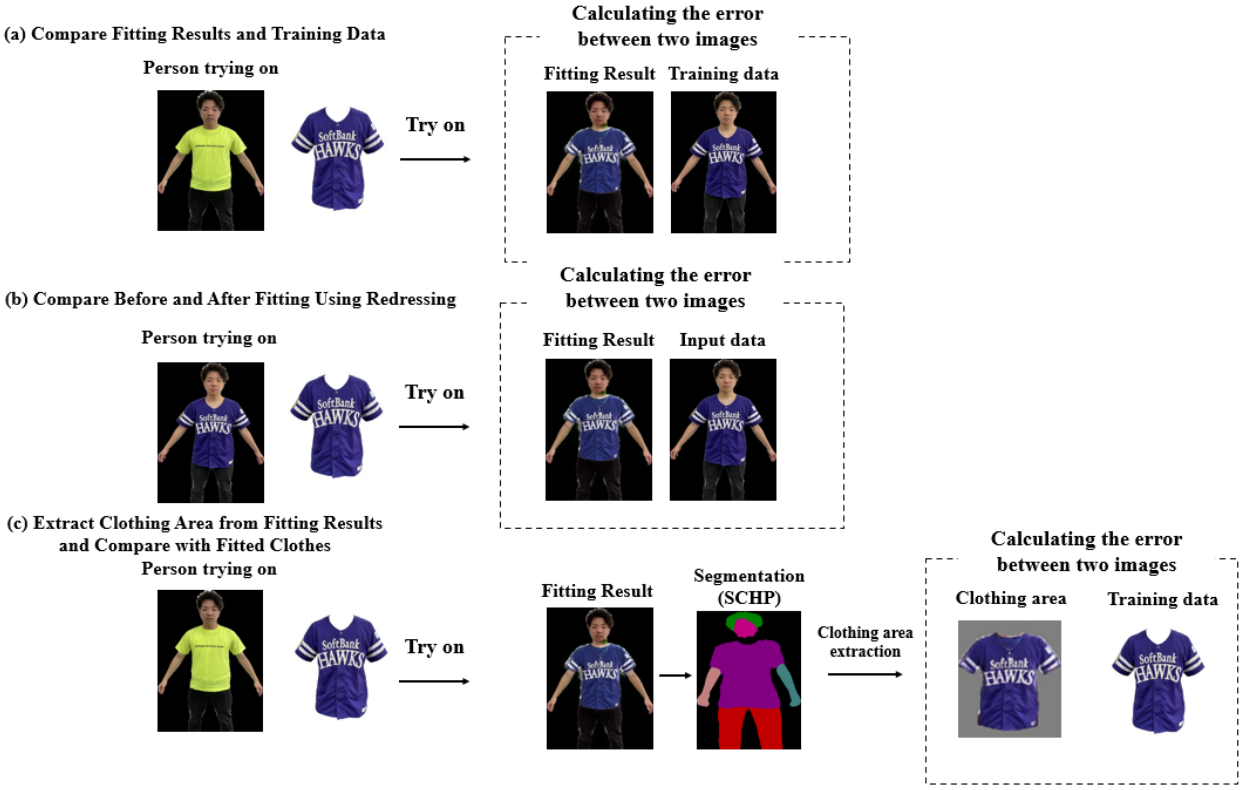

Fig. 3. Calculation of Evaluation Value by SWD

Fig. 4. Decrease in Fitting Accuracy due to Overfitting

V. Proposed Method

Existing methods have issues with data scarcity and the overfitting that results from it. Therefore, in this study, we use a system that employs a genetic algorithm (GA) to control the learning network, aiming to enhance the learning efficiency [7].In previous virtual try-on approaches, all networks within the system had their parameters updated through training. However, due to the limited data in the real-world dataset, as shown in Fig.3, overfitting is more likely to occur in systems with many networks. In contrast, our method can selectively choose networks that improve try-on accuracy through learning by switching the learning of networks using GA. We can enhance try-on accuracy by preventing overfitting and achieving efficient learning with a small amount of data. The networks to be switched for learning are G1, G2 within SGM, the discriminator of G3 within CFM, and U-Net. These networks are selected and trained using GA. After each training, try-on results are generated, and an evaluation value is calculated. By searching for networks with higher evaluation values using GA, we can ensure optimal learning is consistently performed, improving try-on accuracy. We perform transfer learning on a pre-trained model using the real-world dataset in the experiments. The algorithm and the process flow are explained, and the processing flow is shown in Fig.4.

Step1: The networks to be trained are specified using arrays. The learning networks are assigned a value of 1, while the non-learning networks are assigned a value of 0, corresponding to the four networks: G1, G2, G3, and U-Net. The initial population consists of 10 individuals, generated by creating an array of length 4 with random values. The learning networks to be trained are determined based on this population.

Step2: Training is performed based on the generated individuals. A one-epoch training is performed with the conditions of the 10 generated individuals, and the models for each network are saved.

Step3: Fitting images are generated using the saved models. Ten fitting results are generated from the ten saved models, and

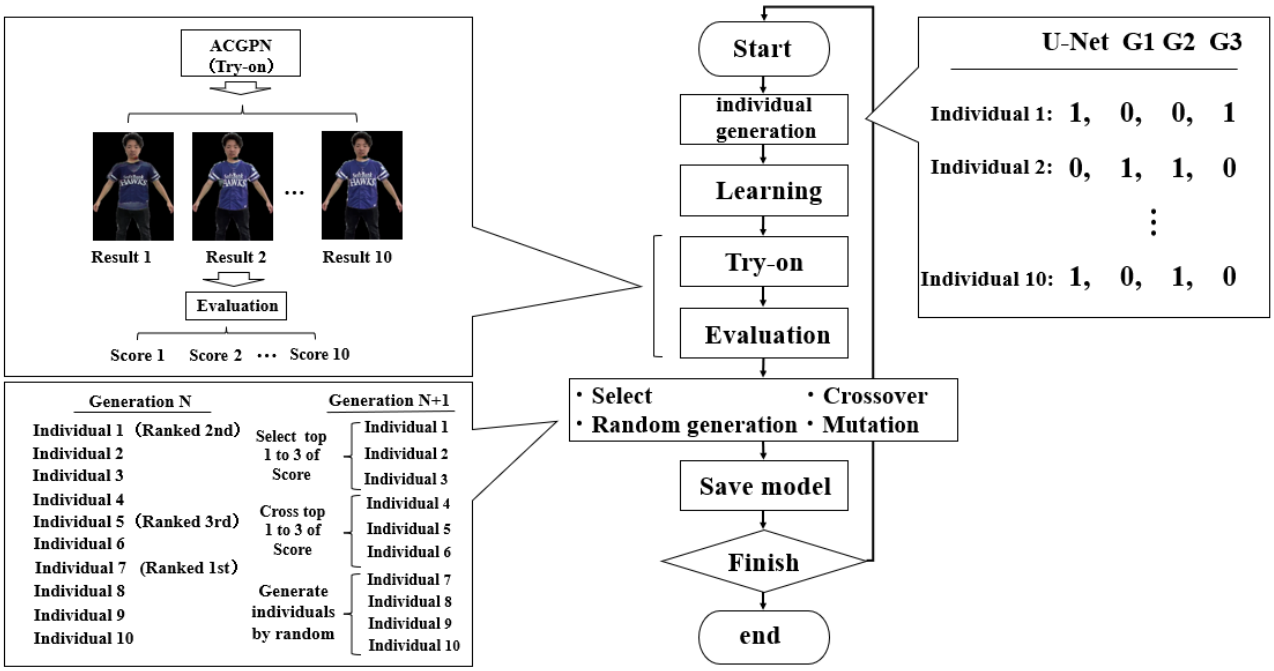

Fig. 5. flow of the Proposed Method

comparisons are made using each evaluation method, followed by evaluation using the SWD.

Step4: Next-generation individuals are generated based on the evaluation values obtained in Step 3. The top three individuals with the highest evaluation values are selected as they are for the next generation. The selected three individuals undergo crossover, generating three new individuals as the next generation. The remaining four individuals are randomly generated to complete the generation of next-generation individuals.

Step5: The model of the individual with the highest evaluation value in the current generation is used for training in the next generation.

Steps 2 to 4 are repeated until the final generation.

VI. Experiment and consideration

In the experiment,We compared the try-on results generated from models trained on the VTON dataset, models trained on the real-world dataset, and models trained using the proposed method. We also compared the evaluation metrics used in the GA.

Fig.6 presents the try-on results obtained from the experiment using four types of input data. In 1 and 2, we compare 10 epoch fine-tuning using a real environment dataset and 10 epoch transfer learning using a real environment dataset using the proposed method (GA).In addition, in 2(a)-(c), we compare the three evaluation metrics used for GA. The training process for each case is as follows: **1**.Fine-tuning with the real-world dataset for 10 epochs using a pre-trained model on the VTON dataset.

2. Transfer learning with the real-world dataset for 10 epochs using a pre-trained model on the VTON dataset, incorporating the proposed method (GA). **(a)**The evaluation metric used in the GA compares between the try-on results and the training data. **(b)**.The evaluation metric used in the GA compares between the pre-try-on and post-try-on results obtained through re-dressing. **(c)**.The evaluation metric used in the GA compares between the extracted clothing region from the try-on results and the try-on clothing.

The try-on results obtained using the proposed method utilized the model with the highest evaluation value among the final generation individuals in the GA.

Fig.6 shows that fitting accuracy improved with any evaluation method compared to the existing method fine-tuning for 10 epochs. We used the VTON dataset and compared the existing method, fine-tuning for 10 epochs on a real-world dataset, with the proposed method. Overfitting tends to occur as the existing method involves training all networks for 10 epochs. overfitting tends to occur. In the proposed method, the learning of the network is controlled by the GA to prevent overfitting. As a result, we confirmed an improvement in fitting accuracy with the same condition of 10 epochs of learning, proving the effectiveness of our proposed method.

The examination of variations under different evaluation methods was conducted. The method of using teacher data for evaluation (a) was presumed to enable the natural fitting of clothes. Compared with the methods using other evaluations, it was confirmed that the colors and logos of the clothing were most well-preserved, thus producing the most superior results. As for (b), because the evaluation was conducted by re-dressing, natural fitting was expected. Consequently, no unnatural fittings were observed. For (c), the clothing area

Fig. 6. Fitting Results from Learning

was extracted and compared with the tried-on clothing, but no significant changes were observed compared with other evaluation methods. Additionally, as a result of comparing the extracted clothing area, it was expected that only the features of the clothing would be preserved, and features other than the face would not be preserved. However, the same results as other test results were obtained. These results concluded that there was no significant difference between any of the evaluation methods used for GA.

VII. CONCLUSIONS AND AFTERWORKS

In this paper, we proposed a system that uses GA to control networks that perform learning, aiming to enhance learning efficiency, improve fitting accuracy, and prevent overfitting due to data scarcity. Through our experiments, we confirmed changes under different evaluation methods, validating the improvement in fitting accuracy and the effectiveness of our system. A current issue is that the evaluation value of GA is calculated from specific training data. Because the training data consist of images not found in existing learning datasets, preparation involves time and costs akin to real-world datasets. Hence, a method that enables evaluation solely on fitting results without needing training data was required. However, our current evaluation method showed that a certain level of fitting accuracy can be achieved even without using training data.

The proposed method has a property in its algorithm where ten individuals per generation learn for one epoch each. As a result, 100 learning epochs took place in the ten generations of the experiment. The amount of learning increases exponentially as generations increase, leading to a massive amount of learning time.Therefore, modifications to the algorithm will be necessary to increase the number of data and the number of learning iterations in the future. We are also considering improvements to human segmentation to enhance fitting accuracy.

REFERENCES

[1] Han, Xintong, et al. “Viton: An image-based virtualtry-on network.” Proceedings of the IEEE conferenceon computer vision and pattern recognition. 2018.
[2] Minar, Matiur Rahman, et al. “Cp-vton+: Clothingshape and texture preserving image-based virtualtry-on.” CVPR Workshops. 2020.
[3] Yang, Han, et al. “Towards photo-realistic virtualtry-on by adaptively generating-preserving imagecontent.” Proceedings of the IEEE/CVF conferenceon computer vision and pattern recognition. 2020.European Journal of Operational Research Vol.28 pp.271–287 2001
[4] Ge, Yuying, et al. “Parser-free virtual try-on viadistilling appearance flows.” Proceedings of theIEEE/CVF conference on computer vision and pattern recognition. 2021.
[5] S. Nagae, S. Kawai and H. Nobuhara, “Transfer Learning Layer Selection Using Genetic Algorithm,”2020 IEEE Congress on Evolutionary Computation (CEC), 2020.
[6] Fernando. C, Banarse. D, Blundell. C, Zwols. Y, Ha. D, Rusu. A, Pritzel.A, Wierstra. D, “Pathnet: evolution channels gradient descent in superneural networks,” arXiv preprint arXiv:1701.08734, 2017.
[7] Taiyo Sato,Takumi Ikenaga,Shudai Ishikawa “Improving the Accuracy of Virtual Try-On with Real-World Dataset Using Genetic Algorithm” THE INSTITUTE OF ELECTRONICS,INFORMATION AND COMMUNICATION ENGINEERS TECHNICAL REPORT OF IEICE.2023.3
[8] Li, Peike, et al. “Self-correction for human parsing.”IEEE Transactions on Pattern Analysis and MachineIntelligence (2020).
[9] Cao, Z., et al. “OpenPose: Realtime Multi-Person2D Pose Estimation Using Part Affinity Fields. InIEEE Transactions on Pattern Analysis and MachineIntelligence.” (2019).
[10] Takumi Ikenaga,Shudai Ishikawa “Creating the natural datasets by the clothing extraction module and it adopts to virtual try-on system” 28th International Symposium on Artificial Life and Robotics 2023

[11] Nadjahi, Kimia and Durmus, Alain and Jacob, Pierre E. and Badeau, Roland and Şimşekli, “Fast Approximation of the Sliced-Wasserstein Distance Using Concentration of Random Projections”