DEEP GENERATIVE MODELS FOR POSE & FASHION TRANSFER

Ho Kang Qi Ethan and Hong Deyu
Hwa Chong Institution (High School), 661 Bukit Timah Rd, Singapore 6468 3955
Ms Wong Minn Xuan and Dr Shen Bing Quan

---

1. Abstract

This paper reviews a deep generative model for controllable image synthesis, specifically the viability of Attribute-Decomposed GAN (ADGAN) proposed by Men et al, that produces realistic person images with flexible human attributes (e.g., pose, head, upper clothes, pants) provided through various source inputs. ADGAN embeds these attributes into the latent space via independent codes in order to be controlled and mixed into desired images. This strategy allows for the user to achieve more precise control over pose and fashion transfer. This paper also attempts to fill in the gaps left by the source code of ADGAN that prevented it from being used, and proposes the usage of an alternative pose estimator, Mediapipe, instead of Open Pose which is used in ADGAN, due to Mediapipe being more precise.

2. Introduction

Person Image Synthesis is extremely crucial in the domain of Computer Vision and Computer Graphics due to its many potential applications such as image editing, fashion transfer, etc. This topic requires image generation to retain its target pose and/or components (e.g. shirt, head etc.) in order to render photo-realistic generated images. In this project, we aim to research the viability of a generative model that specialises in the domain of fashion transfer, and attempt to add our own improvements to it.

3. Theoretical Framework/Reference Models

Generative Adversarial Networks (GAN) have become one of the most powerful generative models for image synthesis in the past few years, with Pix2Pix solving the image-to-image translation task with conditional GANs. CycleGAN attempted to generate images from two domains using unlabelled images. In this section, we focus on two GANs: StyleGAN and ADGAN.

a. StyleGAN

StyleGAN is a generative adversarial network introduced by Nvidia researchers in 2018, and made source available in 2019. It depends on CUDA, a software by Nvidia, GPUs and Google's Tensorflow for its image generation. StyleGAN synthesised images by using a new generator architecture controlling the generator via adaptive instance normalisation. (AdaIN) (a normalisation method that aligns mean and variance of content features with those of style features) However, StyleGAN isn't optimal for controllable pose and clothing transfer since the techniques it uses have limited scalability in handling attribute guided person synthesis.

b. ADGAN

ADGAN is a generative adversarial network proposed by Men Yifang et al in 2020. It was developed as a potential solution to the problems StyleGAN faced. Its generator architecture is designed with attribute decomposition, transferring not only pose of reference image to target person but its various components as well, such as the upper clothes and the pants. It is done through embedding the target pose and source person into the latent space through two separate pathways, called pose encoding and decomposed component encoding. The desired image is then reconstructed via a decoder. This allows for flexible and controllable user control of human attributes, an improvement from its predecessors.

4. Materials & Methods

The GAN model consists of a generator and a discriminator, generating images based on input and distinguishing whether the image is generated or original respectively. Through several epochs of training, the generator and the discriminator will attempt to win against the other, adjusting its weights based on the result of every epoch.

We have decided to refer to ADGAN's model in order to accomplish our objectives. Instead of labelling different data for each attribute, the model achieves automatic and unsupervised separation of component attributes by using a well-designed generator created by ADGAN. Thus, the dataset needed will contain person images with each person in several poses. The corresponding keypoint-based pose can be extracted using an existing pose estimation method. The code for the generator and the discriminator, together with the pose and component decoder, has been provided by ADGAN.

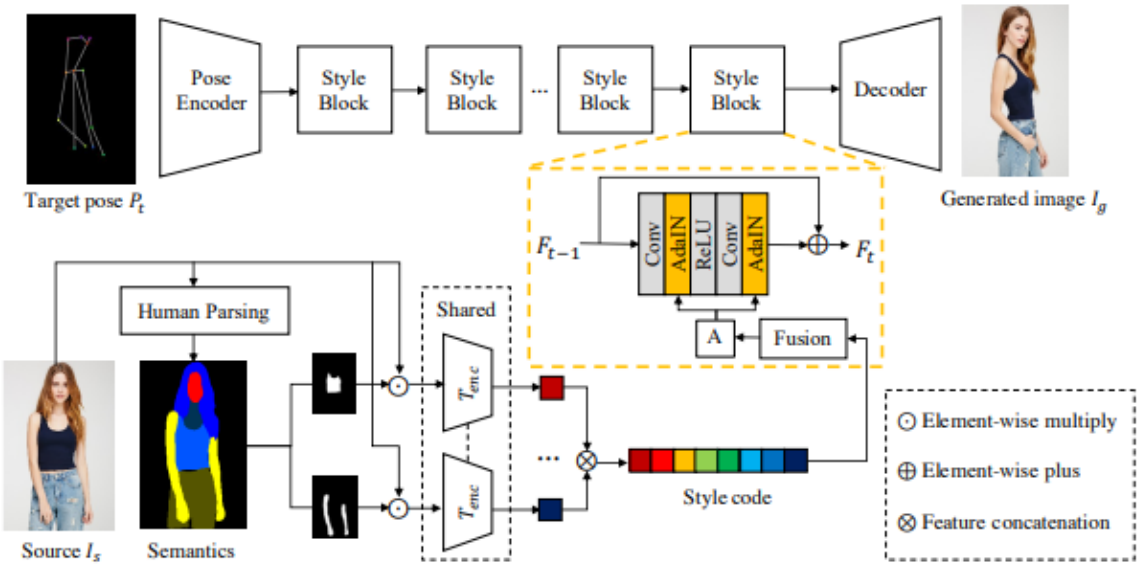


Figure 1: The architecture of the ADGAN network generator. The target image's pose that was fed as input will be extracted from the image and encoded by the pose encoder. The source image's different components will be separated using human parsing, which is then embedded as style code. The style code is then injected into the various style blocks as chosen by the user, before the image is decoded to produce the generated image.

During training, a target pose and a source image are fed into the generator, and the resulting synthesised image, which has the appearance of the source image but instead with the target pose's pose, will be challenged for its realness by the two discriminators of ADGAN.

We have made a multitude of changes to ADGAN's code, file structure and renaming, due to ADGAN's code being incomplete and using outdated versions of certain modules, as such duplicating our ADGAN repository is recommended.

The train and test datasets have already been prepared for testing, together with the pose heatmaps. Since ADGAN's OpenPose pose estimator code returns no results, we have decided to use Mediapipe as an alternative pose estimator to extract pose keypoints.

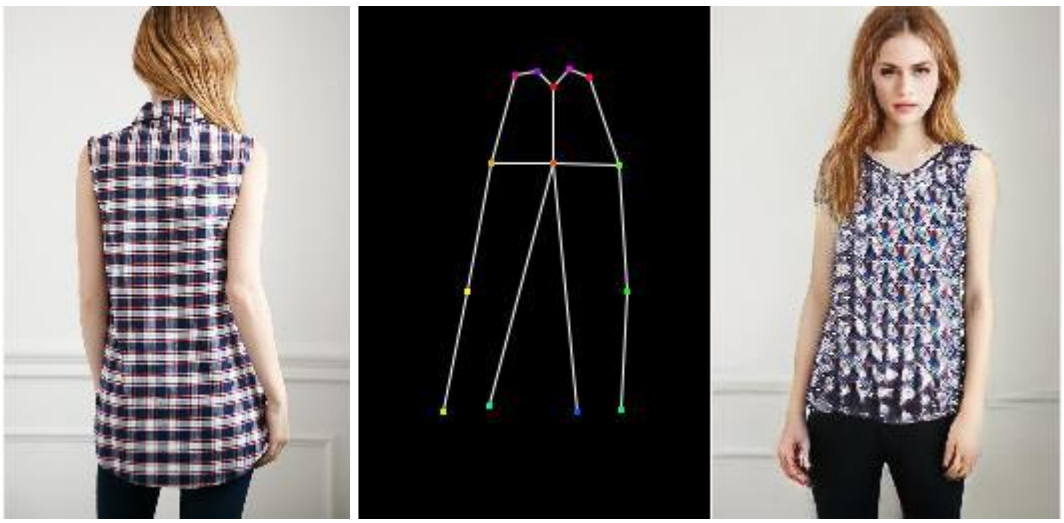
Mediapipe is a cross-platform pipeline framework that builds custom machine learning solutions. For this project, Mediapipe's pose estimation is used. It provides a higher number of keypoints compared to Openpose's, which allows for more precise estimation of pose. However, we have yet to integrate the larger number of keypoints into the model due to ADGAN's model not accepting the higher number of channels.

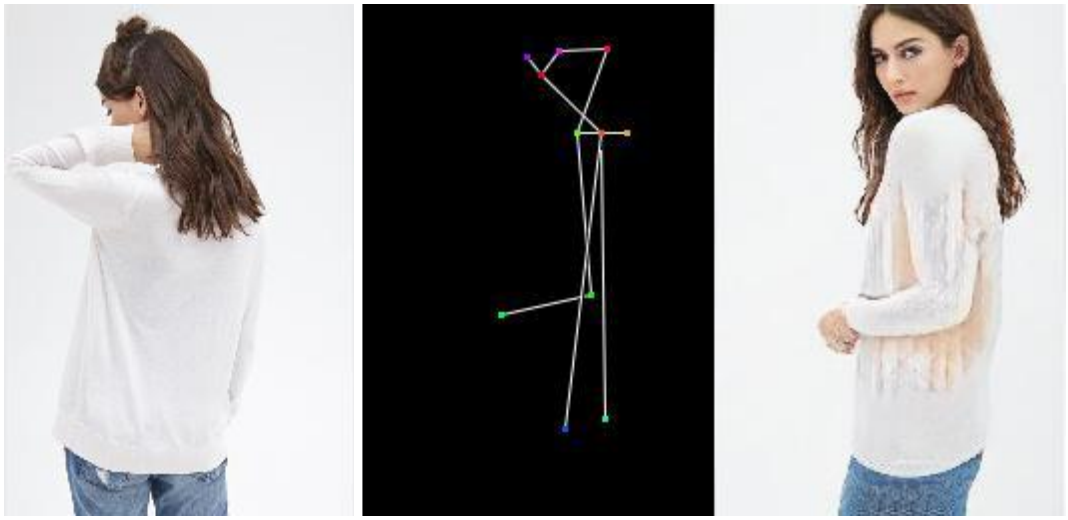
Figure 2: Mediapipe's extraction of pose keypoints.

The two defining components of the network are the pose and attributes. The attributes are encoded using a fixed VGG encoder and a learnable encoder in the generator, while the pose is encoded through a pose encoder. They are then reconstructed using a standard decoder.

Training involves the usage of different losses such as perceptual loss and reconstruction loss to adjust the weights of the generator to produce better results. It uses two Nvidia Tesla-V100 GPUs.

5. Results





Figures 2: The source image, the target pose and the generated pose.

After testing the model, it is revealed that it does indeed generate realistic and natural results in most cases, even when the source image is facing behind as shown in Figures 2 & 3.

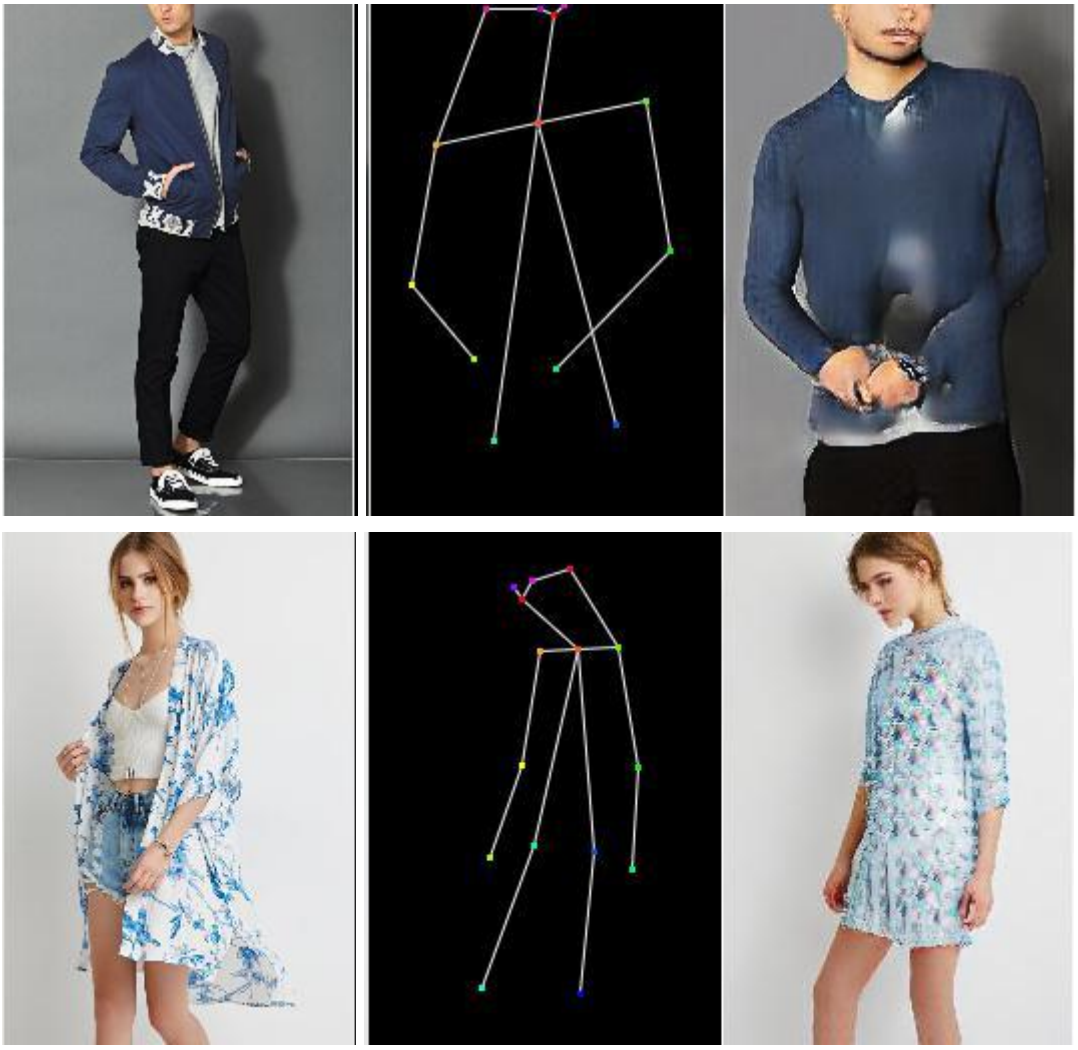


Figure 3: Failure cases where the generator does not produce realistic results of clothes.

However, there are still failure cases where the generator fails to produce a realistic image. These cases are often when the source image has more than two layers of clothing, due to the segmentation encoder not being able to detect attributes such as jackets or coats. As such, it often fuses the outer garment into the top, thus generating an image that does not look realistic.

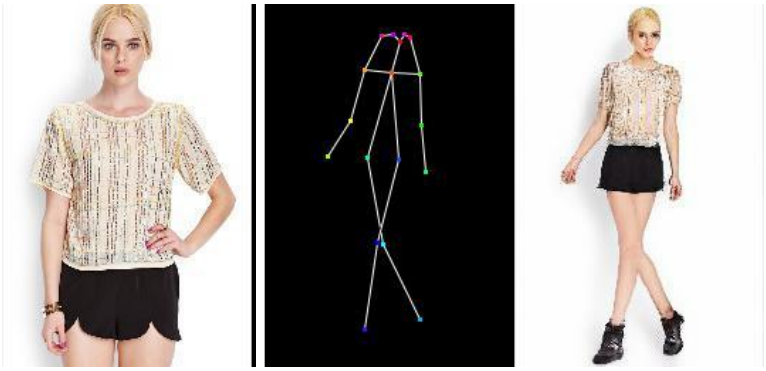


Figure 4: The pattern on clothing is not accurate to the source image

There are also many cases of patterns of clothing not being able to be transferred accurately to the generated image. This could be due to data lost during the encoding decoding process of style code, or the segmentation parser not being precise enough.

We've used the inception score to evaluate the realism and accuracy of the results. The inception score calculated was 3.301 for the batch of results.

Model	**IS**
PG^2	3.202
DPIG	3.323
Def-GAN	3.265
PATN	3.209
ADGAN	3.364
Ours	3.301

Figure 5: Quantitative comparison with state of the art methods

6. Discussion

While ADGAN does produce rather impressive results, there are issues still present in the model such as biases in the model and intricate patterns. However, the biggest problem we faced attempting the project was how the source code was at times incomplete and used outdated modules.

We believe that further tidying up the code of ADGAN would render it much more effective to be usable for people interested in utilising the model, since currently the code requires much debugging in order to even test the model.

Due to this project being the first time we've ever touched code related to Artificial Intelligence, we spent the bulk of our time learning about the architecture and figuring out how it works. As

such, ADGAN's code being not very reader-friendly and instructions being somewhat vague posed a great challenge to us.

However, we've learned a great deal from this project. This is the first time we've ever embarked on a project of this difficulty, and learning skills such as cooperation, problem solving, communication etc was very valuable for us.

7. Acknowledgement

This project is only possible with the work done by Men et al on ADGAN and the developers of Mediapipe, the guidance of our mentors, Dr Shen Bing Quan and Ms Wong Minh Xuan, and the opportunity given by DSO and Hwa Chong Institution.

8. References

[1] Menyifang. (n.d.). *Menyifang/Adgan: The implementation of paper "controllable person image synthesis with attribute-decomposed gan" CVPR 2020 (oral); pose and appearance attributes transfer;* GitHub. Retrieved December 30, 2022, from https://github.com/menyifang/ADGAN

[2] *Live ML anywhere*. MediaPipe. (n.d.). Retrieved December 30, 2022, from https://mediapipe.dev/