Stacked deformable GAN for pose-based person image generation

Lilang Lin[1], Shixing Yu [1] , Sibo Geng[1],
[1] Peking University
{1700017758, yushixing, 1700017802}@pku.edu.cn,

Abstract

In this paper, we introduce a novel method to generate human image with pose information as guidance. Our task is to generate an image that is similar in context(e.g. background, clothing, body build...) with an input image x_a, while of the same pose given by a 2-D points sequence indicating the human-pose's skeleton. To deal with pixel-to-pixel misalignments caused by the pose transformation, we use a 2-stream structure that separates the context information and pose information into two tracks and extract their features respectively. Then the context and pose information are aggregated in different stages to generate images of different scales. To improve the performance of existing method in high resolution, we propose to use a stacked network architecture to generate in a stage-wise method, from low resolution to high resolution. The classical generator-discriminator framework is used in our pipeline and the model is trained by playing the minimax game. We test our algorithm using dataset Market-1501. Qualitative results and ablation studies show the effectiveness of the proposed method.

1. Introduction

Nowadays, person image editing applications are popular around the world, which takes a RGB image of a person as input, and the editor can implement some adjustments on the input image. However, people are not contented with traditional editing methods and functions. Driven by this market demand, researches have been widely done to meet the needs of developing user-friendly functions. Image generation has been a broadly explored topic with the help of convolutional neural networks, especially under the context of person image editing. As the most popular generative model, GANs are leveraged to generate faces and natural scene images[13]. Besides, training and convergence properties are also widely studied to guarantee stability and to improve performance[1].

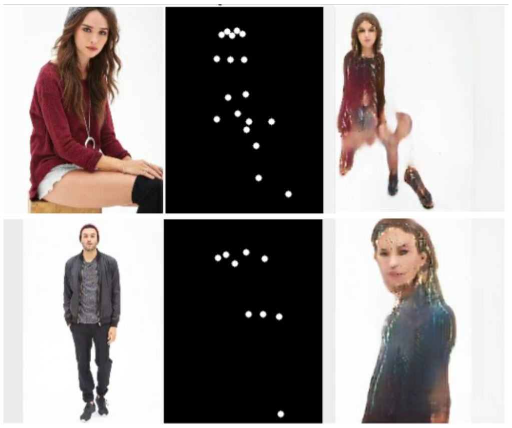


Figure 1. The failure case showed in the Ma's paper[10]. The first row is the original image, the middle row presents the target skeleton and the last row shows the generating result of their method.

Motivated by the interest in synthesizing videos[19] with non-trivial human movements and generating rare poses for human pose estimation[3], pose-based human-being image generation is capturing researchers with its interesting real world application. Inspired by Ma et al.[10]'s work, what we aim to do in this paper is to let the user directly guide the generating process by explicitly editing an appropriate representation of the pose, namely the skeleton of various body joints. The whole process is as follows: Given a full-length person image, extract its pose information, edit the person pose under user's instructions and generate a novel person image which fits the appointed pose.

The task of transferring a person from one pose to another is challenging, previous work addressed this problem with the help of a two-stage GAN[10]. But as we can see in 1, the generating result fails in certain circumstances. We can also see severe artifacts and unrecognizable generated context with bad visual effects. To maintain better generating results, Siarohin et al. [15] propose a deformable GAN by processing pose information and context information (original input image) separately. Later, the two information streams are aggregated by skip-connection. The two task of extracting pose information and extracting context information (person features) are decoupled. Pose transformation is implemented through applying affine transformation on the feature map of the pose heat map.

Although deformable GAN generates more reasonable results, it also presents twisting context and incomplete body parts. To further solve these problems, we are inspired by the idea of StackGAN[21] and designed a stacked network architecture that make use of skipped connection mentioned in deformable GAN. Our contributions are summaried as follow:

- We propose a stacked network structure with multi-scale synthesis process to generate photo-realistic human images.
- We improves the quality of the generated images and stabilizes the training method by jointly approximating multiple distributions under different image resolutions.
- We apply attention module to improve the pose fidelity and retains body parts.

Our code is publicly available. [1]

2. Related Work

2.1. Deep Generative Models

In recent years, deep learning based generating models like generative adversarial networks(GAN)[5] and variational autoencoder (VAE) [8] are widely used to generate vivid images given a specific training dataset representing a set of distributions.

Generative Adversarial Networks have a classic setting of one generator competing with one discriminator, where the discriminator tries to distinguish the generated samples while the generator aims to fool the discriminator by optimizing itself to estimate the true data distribution.

GAN has been a widely studied topic for its strong generating ability[14, 23, 7, 9, **?**]. Tulyakov*et al.* [17] propose an unconditional video generation model based on GAN by dividing the latent variables into content part and motion part. . He *et al.* [6] explore probabilistic video generation using conditional variational autoencoders and utilize the content and motion information.

2.2. Person Image Generation

GANs have achieved superior performance in generating person images. Generally, pose and description (text) are used as important guidance to instruct the generation process.

Ma *et al.* [10] propose to use pose as guidance to generate images through a coarse-to-fine converting process, implemented with a U-Net-like generator. Siarohin *et al.* [15] deforms the generation process to "text-level" and "pose-level", calculated separately and fused together in the way of skip connections. Balakrishnan *et al.* [2] present a modular generative network which separates a scene into various layers and moves the body parts to the desired pose. Neverova *et al.* [11] utilize dense human pose estimation which maps all human pixels of an RGB image to the 3D surface of the human body to obtain more robust representation of the human pose. And inspired from image-to-image translation, Chan *et al.* [4] synthesize high resolution human dancing videos by skeleton-based projection.

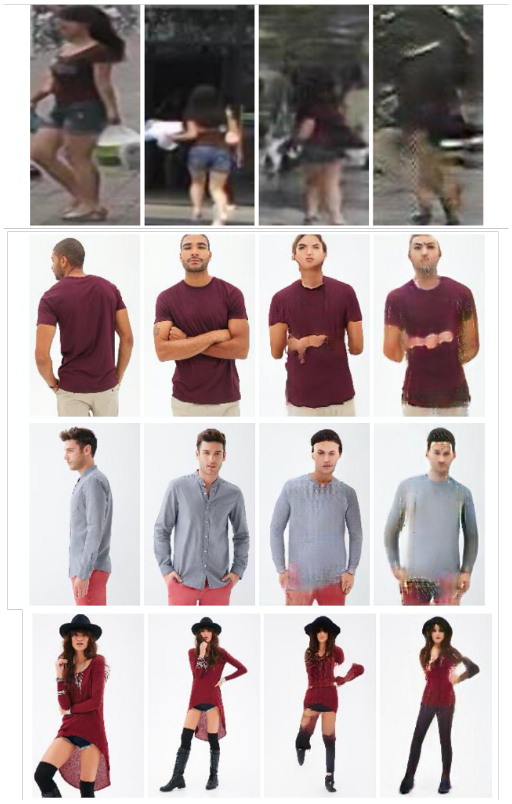


Figure 2. The first row is the original image, the second row presents the ground truth with the specified skeleton, the third row shows the generating result of PIG[10](Pose-based Image Generation), and the last row is the generating result of DGPIG(Deformable GANs for Pose-based Human Image Generation)[15]

Zhang *et al.* [21] utilizes a stacked structure to genreate image from a coarse-to-fine process with a stacked structure consisting of multiple generators and corresponding discriminators. Tao *et al.* [20] employ the attention mechanism into this problem, which is able to synthesize images with fine-grained details from the text. Qiao *et al.* [12] exploits the idea of redescription from cycleGAN [22] and use a semantic text regeneration and alignment module taking the generated image as input to align the given text.

[1] https://github.com/LanglandsLin/SDGAN

3. Our Approach

In this section, we will first describe the method of deformable GAN with two forwarding tracks processing pose information and context information separately. Then, we will describe how we fully utilize context and pose information through a stacked multi-resolution generation process.

3.1. Deformable GAN

If we view different human gestures as different subspaces in the set of all human postures, the deformable GAN section in our model aims to transform the context information from the image input sub-space to the sub-space the skeleton input indicates. We first introduce the procedure our model takes at testing stage, along with some notation. Then methods for training will be addressed. At testing stage, our task will be generating an image $\hat{\mathbf{x}}$ showing a person whose appearance(e.g.,clothes,build) is similar with an input, conditioning image $\mathbf{x_a}$, yet with a body pose similar with the pose $\mathbf{P(x_b)}$. The pose information $\mathbf{P(x_b)} = (\mathbf{p_1}, \mathbf{p_2}, ..., \mathbf{p_k})$ is a sequence of 2-D points indicating the body joints position in the image. $\mathbf{P(x_b)}$ is actually extracted from another image $\mathbf{x_b}$ showing the same person as $\mathbf{x_a}$ but in another posture. Note that the target image $\mathbf{x_b}$ will not be visible to the model during testing time. At training time we use a dataset $\mathbf{X} = \mathbf{x_a^i}, \mathbf{x^i_{b i=1,...,N}}$ containing pairs of images showing the same person in different poses. Pose information extracted from each pair $\mathbf{x_a}, \mathbf{x_b}$ will be denoted as $\mathbf{P(x_a)}, \mathbf{P(x_b)}$. Since connecting low-dimension pose data with high-dimension context data will drastically increase the difficulty for the learning task, a method that represents the pose data with a heat map is proposed by Aiarohin et al.[]. A heat map $\mathbf{H(x)}$ is a 2D matrix of the same shape as the input image, where points closer to the joint position in $\mathbf{P(x)}$ have higher value, while points further away have lower value. The generator $\mathbf{G}$ is fed with: (1) a noise vector z and (2) a tuple $(\mathbf{x_a}, \mathbf{H_a}, \mathbf{H_b})$ containing the original image and heat map and the target heat map. One problem of mapping the context information from the original pose (indicated by $\mathbf{H_a}$) to the target pose (indicated by $\mathbf{H_b}$) via convolution neural network is that, if the differences between the two poses are relatively large(e.g. one facing forward while the other faces backward) the receptive field of the convolution units may not be able to cover the corresponding position in target pose. In other words, context information for target image that may exist in input image can be lost due to long transfer range. Therefore, it would be best to treat context information $\mathbf{x_a}$ and the corresponding pose information $\mathbf{H_b}$ with separate convolution networks. Specifically, $\mathbf{x_a}$ and $\mathbf{H_a}$ are concatenated and processed using stacked layers of convolution units that forms an encoder, while $\mathbf{H_b}$ is processed by another set of convolution units that does not share weights with the encoder that process $\mathbf{x_a}$ and $\mathbf{H_a}$. In order to transfer the context information from the feature maps given by the first stream to the to the sub-space given by the target pose, a pose-guided affine transformation is performed. By comparing the body joints data $\mathbf{P(x_a)}$ and $\mathbf{P(x_b)}$, a local transfer 'mapping'(e.g. rotation, scaling, shifting) can be conducted that is used to 'shuttle' the context information provided by the feature map of $\mathbf{x_a}$ to the corresponding positions at target picture. This process may seem rather procrustean considering that we simply copy and paste the context information from one picture to another. But note that, first, this skipped transformation is performed at feature level, we are only transforming highly encoded latent information between the two pictures, and by the assumption that the real world should be somewhat consistent, this transformation is apt to work. Secondly, the skipped connections are only restricted to a localized area around the joint points, even if the 'mapping' acquired from pose information is not precisely accurate, the affect is very much restricted within the area.

Computing a set of affine transformations. Let $\mathbf{R} = \mathbf{R_1}, \mathbf{R_2}, ..., \mathbf{R_h}$ denote the major body limbs between joints(e.g. the forearm between elbow and wrist), that should be kept intact during the transformation. Let $\mathbf{P^i} = \mathbf{p_1^i}, \mathbf{p_2^i}, \mathbf{p_3^i}, \mathbf{p_4^i}$ be the e rectangular area surrounding a certain limb $\mathbf{R_i}$ in $\mathbf{H_a}$, $\mathbf{Q^i} = \mathbf{q_1^i}, \mathbf{q_2^i}, \mathbf{q_3^i}, \mathbf{q_4^i}$ be the corresponding rectangular area in $\mathbf{H_b}$. Let $\mathbf{f_h}(\Delta; \mathbf{K_h})$ denote the transformation we apply on the feature map. The parameters for the affine transformation can be calculated using Least Squares Error:

$$\mathbf{min_{K_h} \Sigma_j ||q_j - f_h(p_j; k_h)||_2^2} \quad (1)$$

Note that no matter which scale or resolution the output image is, the parameters for the affine transformation can be easily adapted by scaling the original parameters accordingly. With the help of this affine transformation, we are able to transform the context information within a small range of human body limbs to the target image. As for the background contest, another affine transformation calculated by treating $\mathbf{H_a}$ $\mathbf{H_b}$ as 'body limbs' described above. This transformation can be used to shuttle texture information for back ground pixels, to fill in the blanks.

3.2. Stacked network architecture

Compared to a high-resolution image, a low-resolution one is an overview of the image's content. In analogy of human artist's painting process, a draft is often drawn at the beginning and details are added in the following step. The same methodology can also be applied in the image generation task.

Intuitively, a low-resolution image doesn't have enough pixels thus detailed information are often blurred, while we can focus on depicting the object shapes, spatial locations and rough colors. For generative models, the support of

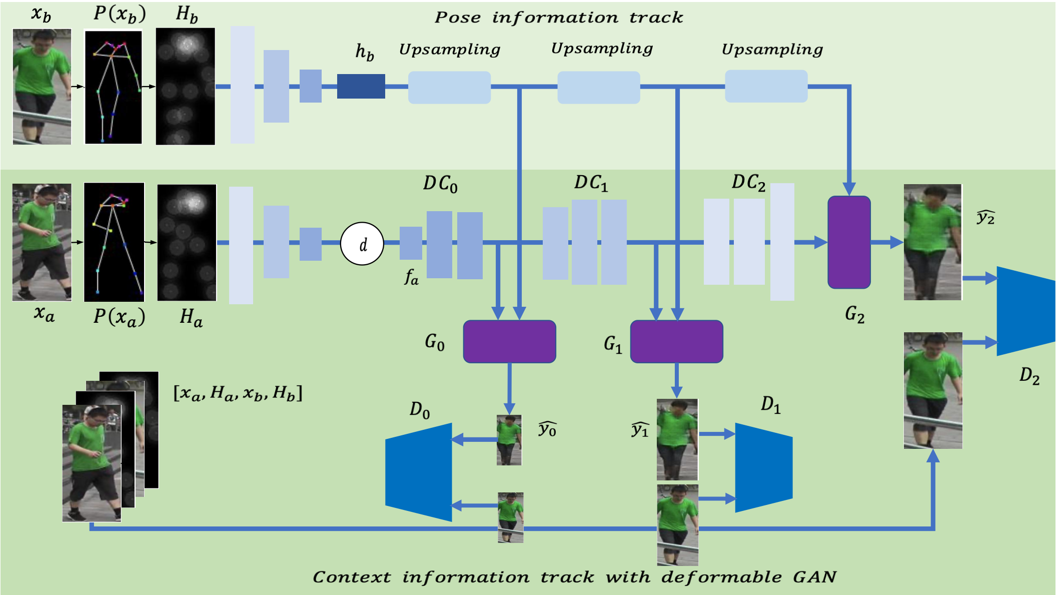


Figure 3. Our network architecture consists of two parts. The track on the above is the pose information track which encodes the pose heat map H_b to a latent code h_b. The track below is the stacked low-to-high-resolution generation process with generator and discriminator at each scale.

model distribution generated from a roughly aligned low-resolution image has better probability of intersecting with the support of image distribution, which lightens the burden of the deformable GAN structure we mentioned above. In [15], latent variables calculated by input image x_a and pose information H_b are skip-connected and directly sent to generate the full-resolution result, which cannot guarantee a stable output without the refine process.

To solve this problem, we propose to fully utilize the two tracks of information in a stage-wise manner. At each scale i, the generator G_i captures the image distribution of scale i by taking both the context information from the image input sub-space and the sub-space of the skeleton input, while the discriminator D_i estimates the probability that a input image came from dataset distribution of that scale rather than the a synthesized one from the generator. Specifically, the first decoder DC_0 takes the feature map after the affine transformation f_a as input, outputs a feature map with $\frac{1}{2^{n-1}}$ scale of the final image l_0, where n is the total number of the decoder.

To fit pose information into the scale 0 and fully utilize it, we pass h_b, the latent code of H_b, through an up-sampling layer to produce another feature map with the same resolution of the output of DC_0. After that, we aggregate the decoded transformed latent variable with pose-based feature map and forward them into the generator to synthesize the image with target pose at scale 0 $\hat{y_0}$.

The process can be formulated as follows:

$$l_0 = DC_0(f_a) \tag{2}$$

$$\hat{y_0} = G_0(l_0, Upsampling_0(h_b)) \tag{3}$$

Low-resolution images generated in scale 0 usually lack vivid object parts and might contain shape distortions and details of the human body parts can be omitted (results will be showed in ablation studies). But general contours, object and color distributions are estimated more properly and is apt to be optimized to approximate a low-resolution version of the final result. In the following stage, generation results are conditioned on low-resolution image representations and the corresponding pose-based features. The decoder will try to excavate high frequency information so that the generator of the same scale can be capable of completing previously ignored information and synthesizing more photo-realistic details.

So in every layer that follows, the decoder at scale i will takes the latent variable of the previous scale l_{i-1} as input. Scale i's latent variable l_i will then aggregate with the up-sampled latent code h_b and the generator will synthesize an image with higher result. The whole generation process can be presented as

$$l_i = DC_i(l_{i-1}), i = 1, 2, ..., n \tag{4}$$

$$\hat{y_i} = G_i(l_i, Upsampling_i(h_b)), i = 1, 2, .., n \tag{5}$$

where $Upsampling_i$ is the upsampling layer that rescales the latent code h_b so that it fits into the target resolution.

Following each generator G_i, a discriminator D_i will distinguish the generated images from the real images sampled from the dataset distribution. The generative adversarial loss is then added as a constraint at each scale:

$$\mathbb{L}_{D_i} = -\mathbb{E}_{x_i \sim p_{data_i}}[logD_i(x_i)] - \mathbb{E}_{z_i \sim p_{G_i}}[log(1 - D_i(z_i)] \quad (6)$$

where p_{data_i} is the dataset distribution while p_{G_i} is the distribution estimated by the generator at scale i.

3.3. Implementation details

The encoders are mainly consist of 3×3 feature extraction convolutions and 4×4 convolutions with stride 2 as downsampling method. Deconvolutions are used in the decoder implemented by PyTorch module ConvTranspose2d to perform upsampling. Leaky ReLU is used in the decoder as activation function as Leaky relu helps to make sure the gradient can flow through the entire architecture. That's an important consideration for any machine learning model, but even more important for GAN's.

Instace normalization[18] and dropout[16] are applied to accelerate convergence of the generative model.

4. Experiments

4.1. Datasets

The Market-1501 dataset contains 32,668 images of 1,501 persons captured from 6 different cameras, which are low-resolution images (128×64) and have high diversity in pose, illumination, background and viewpoint. To train our model, we need pairs of images of the same person in two different poses. As this dataset is relatively noisy, we use 263,631 training pairs which are clear to recognize. For testing, we randomly select 12,000 pairs. No person is in common between the training and the test split.

4.2. Training Details

Following the setting of [15], we train generator and discriminator for 20 iterations, with the Adam optimizer . We also use instance normalization to achieve better performance. And we train each scale synthesized image with L1 loss to give more strict constraint on the shape and location of the images.

We train the network on one NVIDIA Titan X GPU with a mini-batch size of 4 and the learning rate declines from 0.01 to 0.0001 with 0.1 decay rate for every 100 iterations.

4.3. Comparison

In this section, we compare our method stacked deformable GAN (SD GAN) with [15]. To give a comprehensive and thorough evaluation, we conduct experiments under different settings, including baseline, deformable GAN and our method.

Baseline: We use the standard U-Net architecture without deformable skip connections. The inputs of G and D and the way pose information is represented is the same as [15]. However, the encoder of G is composed of only one stream.

Deformable GAN: This is the full-pipeline in [15]. We train this model using an L1 loss together with the adversarial loss.

SD GAN: It is our full system. To obtain high-resolution images, our method generates low-resolution image first. Particularly, SD GAN generates three images every time, and we regard them as low-resolution, middle-resolution and high-resolution, respectively.

In Fig. 4 we show some qualitative results. These figures show the improvement through the baseline. For each result, columns 1, 2 represent the input, column 3 represents the ground truth and column 4 represents the output.

In fact, while pose information is usually well generated by all the methods, the texture generated by baseline often does not correspond to the texture and is blurred. Our method improve much compared with the baseline and can be comparable to Deformable GAN. However, the stacked structure seems not to bring much improvement. We analyze it may be caused by two reasons. On the one hand, the dataest contains only low-resolution images. When we down-sample those images, generating those images can not bring much information. On the other hand, the features we use to generate low-resolution images may not be efficient for the final generation.

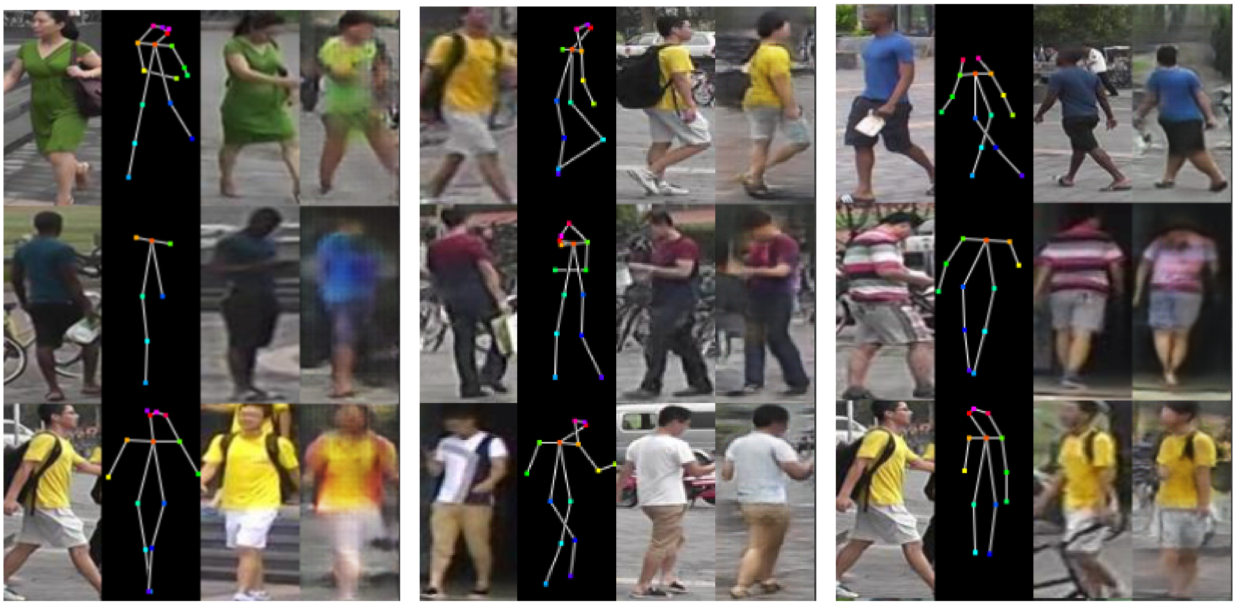
4.4. Ablation Study

Next, we conduct ablation experiments to give more analysis of our proposed approach. All the ablation studies are performed on the Market-1501 dataset.

In fig. 5, we show the different scales of images. The low-resolution generated image is blurred and contains many wrong areas, while the middle-resolution generated image fixes the mistake and shows the fundamental layout according to the skeleton. Based on the framework we generate, our model enhances the details and obtain the final output.

5. Conclusion

We propose a novel upscale module named SD GAN to solve the pose-guided generation task. The staged generating process enabled our model to focus on context transformation and detail refinement respectively, thus acquired better performance. Reviewing our work, the major novelty is demonstrated by the combination of the idea of skip-connection using affine transformation and that of staged-generation process. By transforming feature maps of different scales using the same set of mapping rules, we are



(a) Baseline

(b) Deformable GAN

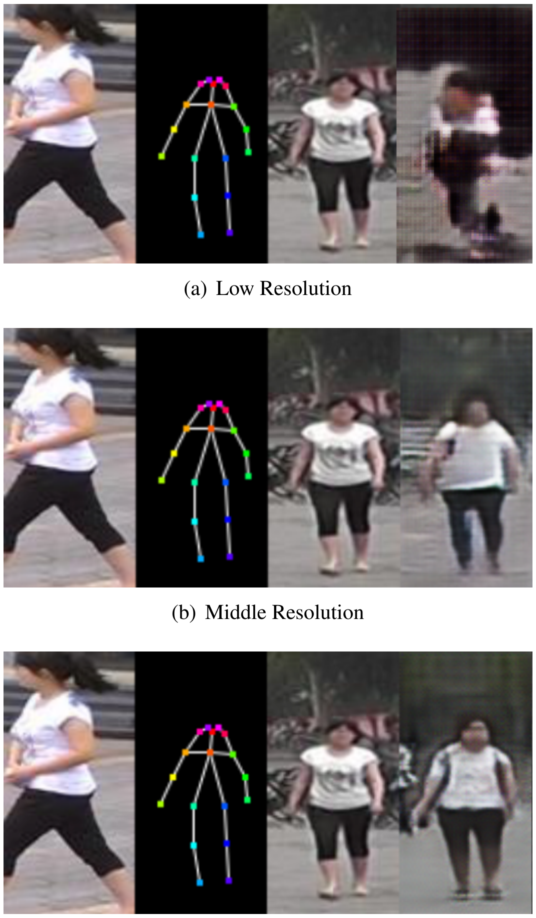
(c) SD GAN

Figure 4. Qualitative results on the Market-1501 dataset.

able to generate images of different resolutions. By applying different loss function that represent different criteria for the generated images, we are able to build a model that not only produces plausible human pictures, but also preserve as much detail as possible.

6. Author contributions

All three of the authors, namely Lilang Lin, Shixing Yu, and Sibo Geng participated in the surveying stage of the project, the architecture of our model is also the result of our joint discussion. Lilang Lin ran the experiments and wrote the experiment section of the report. Shixing Yu and Sibo Geng gave theoretical explanation, post-experiment analysis and wrote other sections of the report.



(a) Low Resolution

(b) Middle Resolution

(c) High Resolution

Figure 5. Different resolution results.

References

[1] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein gan, 2017.

[2] G. Balakrishnan, A. Zhao, A. V. Dalca, F. Durand, and J. Guttag. Synthesizing images of humans in unseen poses. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[3] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[4] C. Chan, S. Ginosar, T. Zhou, and A. A. Efros. Everybody dance now. *arXiv: Graphics*, 2018.

[5] I. Goodfellow, J. Pougetabadie, M. Mirza, B. Xu, D. Warde-farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. pages 2672–2680, 2014.

[6] J. He, A. M. Lehrmann, J. Marino, G. Mori, and L. Sigal. Probabilistic video generation using holistic attribute control. pages 466–483, 2018.

[7] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks, 2016.

[8] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv: Machine Learning*, 2013.

[9] M.-Y. Liu, T. Breuel, and J. Kautz. Unsupervised image-to-image translation networks, 2017.

[10] L. Ma, X. Jia, Q. Sun, B. Schiele, T. Tuytelaars, and L. Van Gool. Pose guided person image generation. pages 406–416, 2017.

[11] N. Neverova, R. A. Guler, and I. Kokkinos. Dense pose transfer. pages 128–143, 2018.

[12] T. Qiao, J. Zhang, D. Xu, and D. Tao. Mirrorgan: Learning text-to-image generation by redescription. pages 1505–1514, 2019.

[13] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks, 2015.

[14] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR*, 2016.

[15] A. Siarohin, E. Sangineto, S. Lathuiliere, and N. Sebe. Deformable gans for pose-based human image generation. pages 3408–3416, 2018.

[16] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014.

[17] S. Tulyakov, M.-Y. Liu, X. Yang, and J. Kautz. Mocogan: Decomposing motion and content for video generation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[18] D. Ulyanov, A. Vedaldi, and V. Lempitsky. Instance normalization: The missing ingredient for fast stylization, 2016.

[19] J. Walker, K. Marino, A. Gupta, and M. Hebert. The pose knows: Video forecasting by generating pose futures. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 3352–3361, 2017.

[20] T. Xu, P. Zhang, Q. Huang, H. Zhang, Z. Gan, X. Huang, and X. He. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. pages 1316–1324, 2018.

[21] H. Zhang, T. Xu, and H. Li. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. pages 5908–5916, 2017.

[22] J. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. pages 2242–2251, 2017.

[23] J.-Y. Zhu, P. Krähenbühl, E. Shechtman, and A. A. Efros. Generative visual manipulation on the natural image manifold, 2016.