Generative Model for Human Pose Transferring between Videos

Shuang Li
CSAIL
MIT
lishuang@mit.edu

Wangzhi Dai
CSAIL
MIT
wzhdai@mit.edu

Zishen Wan
CSAIL
MIT
zishenwa@mit.edu

Abstract

In this project, we performed a human pose transferring between two videos by using generative models. To accelerate the training process of the generative model, we used the low resolution images as input and output and utilize a coarse-to-fine training process to improve the generated results. Quantitative comparison based on Intersection over Union between the generated pose and extracted pose showed this coarse to fine process improve the performance by 10%*. We further proposed a face transfer method from the ground truth images to the generated images, which solved the detail loss in the generated face.*

1. Introduction

Generative models have many important applications in real-world scenarios. In this project, we propose to use generative models to generate motions of movie characters and achieve motion transfer between human subjects in different videos. For example, we can use person in a target video as templates and ask the movie characters in the source video to copy the template's actions.

We need to learn a mapping between images of the two individuals in order to transfer motion between two video subjects in a frame-by-frame manner. And observing that keypoint-based pose, which inherently encodes body position can serve as an intermediate representation between any two object, we design our intermediate representation to be pose stick figures. In order to achieve better transfer effect on face, we propose an face transfer method to directly transfer the face from target person to source person without pose extraction procedure. Furthermore, we cast our video-to-video synthesis part as a distribution matching problem. Given input videos, the goal is to train a model which the conditional distribution of the synthesized videos resembles that of real videos. In this way, we can learn a conditional generative adversarial model given input and output video pairs. With carefully-designed generators and discriminators, we aim to synthesize a protorealistic and temporally coherent video.

Tentatively, we plan to adopt the following steps to achieve this goal. First, we plan to use pose estimation method to extract the motion of a person, and represent it by pose stick figures. Then we use the extracted pose in the template video and apply it to one movie character using a pose transfer model. The pose transfer model is trained to learn the relations between our body and movie characters, so that each human body joint can be well aligned from one person to another. Finally, we generate new actions of the character and make a new movie fragment, according to our aspirations. We use the generative model which take the character image and our pose as inputs and output a new frame with the character having the poses in target video.

2. Related Work

2.1. General Adversarial Networks

Due to advances in image generation and substantial work on general image mapping framework, we can learn a mapping from pose to target subject. With the emergence of Generative Adversarial Networks (GANs) for approximating generative models [1], GANs gradually has many applications including image production [2], mainly because it can generate high quality images with sharp details [3]. Dur-

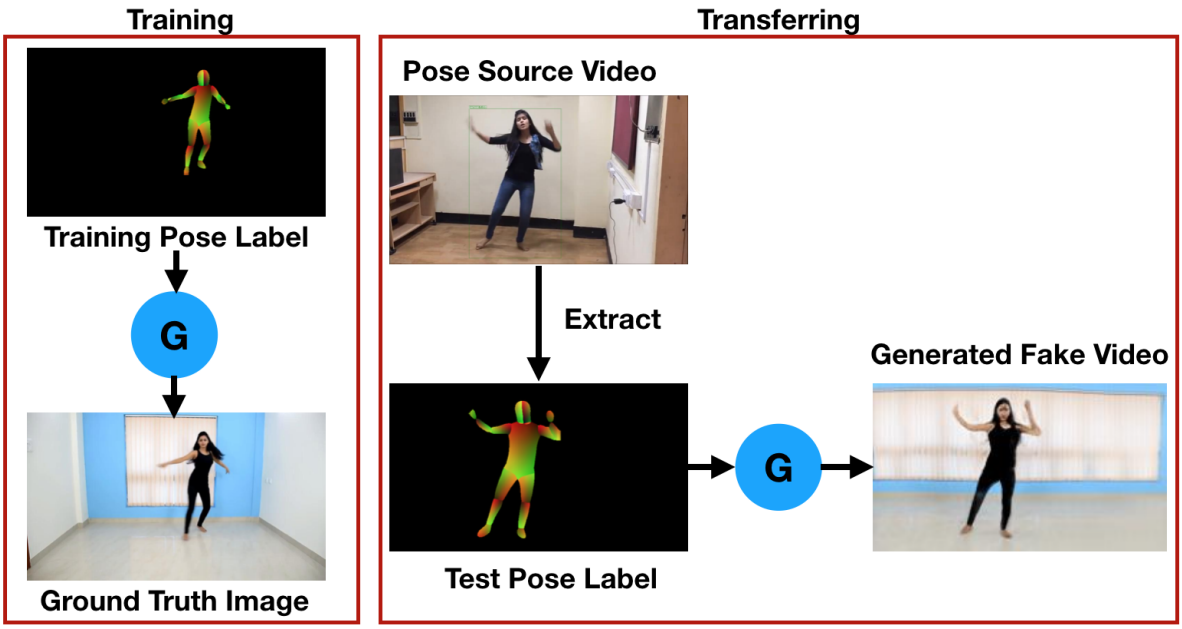

Figure 1. Framework and work flow of the model. The left side shows the training of the generative model from the label to the ground truth images. The right side shows the pose transferring from a source video to the target video using the generative model trained.

ing GAN training, a generator and a discriminator play a zero-sum game. The generator aims to produce realistic generated image so that the discriminator cannot differentiate between real and generated image. Over the past few years there have been several framework, one of which is Conditional GANs, in which the generated output is conditioned on a structured input [4]. Our project belongs to the category of conditional GANs. However, we synthesize photorealistic videos not conditioning on the current observed frames, but conditioning on the extracted poses from source video.

2.2. Motion Transfer

There has been extensive study dedicated on motion transfer. Some methods focused on creating new content by manipulating existing video footage [5], and some approaches using 3D transfer motion for graphics and animation purposes [6]. Recently, Villegas et al. [7] apply deep learning techniques to retarget motion without supervised data. Cheung et al. come up with an elaborate multi-view system to calibrate a personalized kinematic model and render images of human subject performing new motions. In contrast, Caroline Chan et al. [8] explore motion transfer between 2D video subjects where there is a lack of 3D information. Avoiding both source-target data calibration and lifting into 3D space, they achieve per-frame image-to-image translation with spatio-temporal smoothing and dance transfer between different people.

2.3. Video-to-video Synthesis

There are some existed video-to-video synthesis method. Some of exiting approaches rely on the special cases of synthesis problem, such as video super-resolution [9, 10], video matting and blending[11, 12], and video inpainting[13]. Video style transfer [14, 15, 16, 17] is also related, which transfer the style of a reference painting to a natural scene video. Recently, Ting-Chun Wang et al.[18] proposed a video-to-video method outperforming a strong baseline that combines a video style transfer and a state-of-art image-to-image translation approach.

3. Model Overview

Given a video of a source person and another of a target person, our goal is to achieve pose transfer. That is we generate a new video of the person looking like the source person but enacting the same motions as the target. To perform this task, we divide our pipeline into three stages: pose extraction, pose nor-

Figure 2. Original image

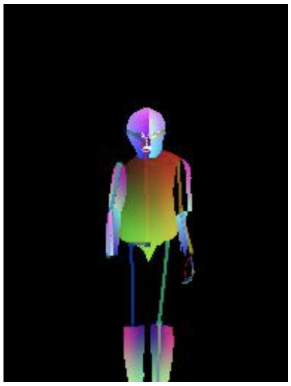

Figure 3. Extracted pose

malization and mapping from pose stick figures to the source subject. In the pose detection stage, we use a pretrained state-of-art pose detector (e.g Densepose) to extract pose stick figures from the source person. In the pose normalization stage, we deal with the differences between the source and target person due to different body shapes and locations in each frame. In the training stage, we design a system to map from the extracted poses to the source images using generative adversarial network.

4. Pose Extraction and Normalization

4.1. Pose Extraction

We use a state of the art pose detector P to create images which can encode body position. The pose detector can accurately estimates 2D coordinates. By plotting the keypoints and drawing lines between connected joints, we draw a representation of the resulting extracted pose as Figure 3 shows.

The pipeline of pose detection stage is as follows[19]. First, we analyze the image to generate a set of feature maps that are the input to the pose detector by using a convolution network . Then, we use a feed-forward network to simultaneously predict a set of 2D body location confidence maps S and a set of 2D vector fields L of part affinities, which encode the degree of association between parts. The set $S = (S_1, S_2, \ldots, S_J)$ has J confidence maps. The set $L = (L_1, L_2, \ldots, L_C)$ has C vector fields, each vector corresponding to each limb. Each branch is an iterative prediction architecture which refines the predictions over successive stages. Finally, the affinity fields and the confidence are parsed by greedy inference to output 2D locations of keypoints in the image.

In the refinement stage, we refine the confidence maps and affinity fields and apply two loss functions for each one to guide the network to do iterative prediction. We adopt an L_2 loss between the estimated prediction and the ground truth maps and fields:

$$f_S^t = \sum_{j=1}^{J} \sum_{p} W(p) \parallel S_j^t(p) - S_j^*(p) \parallel_2^2 \qquad (1)$$

$$f_L^t = \sum_{c=1}^{C} \sum_{p} W(p) \parallel L_c^t(p) - L_c^*(p) \parallel_2^2 \qquad (2)$$

where S_j^* is the ground truth confidence map, L_c^* is the ground truth affinity vector field. The intermediate supervision at each stage addresses the vanishing gradient problem by replenishing the gradient periodically. Therefore, the overall objective is

$$f = \sum_{t=1}^{T} (f_S^t + f_L^t) \qquad (3)$$

4.2. Pose Normalization

In different videos, subjects may have different limb proportions or stand closer or farther to the camera than one another. Therefore when transferring motion between two subjects, it may be necessary to transform the pose keypoints of the target person so that they appear in accordance with the source persons body shape and proportion.

To perform a translation between the detected pose to the targeting video, we have to take into account the difference of character's limb size and their distance to the camera. This asks us to find a mapping between two poses with different sizes. At this stage

of the work, we propose to adopt the linear mapping method as described in [8]. Concretely, in each image, the maximum and minimum foot positions are first found to align a person's position in the scene. Then a transformation is made the locate the source pose's position into the target pose. Given an average ankle position a_{source} in the a source frame, the transformation b is calculated for that frame according to the following equation,

$$b = t_{min} + \frac{a_{source} - s_{min}}{s_{max} - s_{min}}(t_{max} - tmin) - f_{source} \tag{4}$$

where t_{min} and t_{max} are the minimum and maximum ankle positions in the target image and s_{min} and s_{max} are the positions in the source image.

After mapping the location of the foot, we then need to map the scale of the two poses. To calculate the scale, we plan to cluster the heights around the minimum ankle position and the maximum ankle position and find the maximum height for each cluster for each video. Call these maximum heights t_{close} for the maximum of the cluster near the target persons maximum ankle position, t_{far} for the maximum of the cluster near the target persons minimum ankle position, and s_{close} and s_{far} respectively. We obtain the close ratio by taking the ratio between the targets close height and the sources close height, and similarly for the far ratio. Given average ankle position a_{source}, the scale for this frame is interpolated between these two ratios in the same way as the translation is interpolated as described in the following equation,

$$scale = \frac{t_{far}}{s_{far}} + \frac{a_{source} - a_{min}}{s_{max} - s_{min}}(\frac{t_{close}}{s_{close}} - \frac{t_{far}}{s_{far}}) \tag{5}$$

5. Video to Video Synthesis

5.1. Set up

Let $x_1^T = \{x_1, x_2, \ldots, x_T\}$ be a sequence of extracted pose from the source video frames. Let $y_1^T = \{y_1, y_2, \ldots, y_T\}$ be the sequence of corresponding real images. Our goal is to learn a mapping function which can convert x_1^T to a sequence of output video frames, $G(x)_1^T = G(x)_1, G(x)_2, \ldots, G(x)_T$, so that the conditional distribution of $G(x)_1^T$ given x_1^T is identical to the conditional distribution of y_1^T given x_1^T. That is

$$p(G(x)_1^T | x_1^T) = p(y_1^T | x_1^T) \tag{6}$$

Through matching the above conditional distributions, the model can learn to generate photorealistic and temporally coherent output sequences, which seems to be captured by a video camera.

For the conditional distribution matching task, we adopt a conditional generative adversarial network framework. Let G be a generator that can map an input source image sequence to a corresponding output image sequence: $G(x_1^T)$. We train the generator by solving the minimize-maximize optimization problem given by

$$\begin{aligned} \max_D \min_G & E_{y_1^T, x_1^T}[\log D(y_1^T, x_1^T)] \\ & + E_{x_1^T}[\log(1 - D(G(x_1^T), x_1^T))] \end{aligned} \tag{7}$$

where D is the discriminator and G is the generator. To solve this equation, we should minimize the Jensen-Shannon divergence between $p(G(x)_1^T | x_1^T)$ and $p(y_1^T | x_1^T)$ [20].

5.2. Sequential generator

To simplify this video-to-video synthesis problem, we make a Markov assumption where we factorize the conditional distribution $p(G(x)_1^T | x_1^T))$ to a product form as follows:

$$p(G(x)_1^T | x_1^T) = \prod_{t=1}^{T} p(G(x)_1^T | G(x)_{t-L}^{t-1}, x_{t-L}^t) \tag{8}$$

which means we assume the video frames can be generated sequentially. The generation of the t-th frame $G(x)_t$ only depends on three input: a)current source extracted poses x_t, b) past L source extracted poses x_{t-L}^{t-1}, and c) past L generated images $G(x)_{t-L}^{t-1}$. In order to model the conditional distribution $p(G(x)_1^T | G(x)_{t-L}^{t-1}, x_{t-L}^t)$, we train a feed-forward network Q using $G(x)_1^T = Q(G(x)_{t-L}^{t-1}, x_{t-L}^t)$. By applying the function Q in a recursive manner, we obtain the final output $G(x)_1^T$.

Considering the video contains much redundant information, we can use the optical frame to warp the current frame to generate an estimated next frame, if the optical frame from the current frame to the next frame is known. Except for some occluding areas, this estimation process would be largely correct. We model Q based on the observation:

$$Q(G(x)_{t-L}^{t-1}, x_{t-L}^t) = m_t \odot h(t) + (1 - m_t) w_{t-1}(G(x)_{t-1}) \tag{9}$$

The definitions of Equation (9) are:

1: an image with all ones.

$\odot$: the element-wise product operator.

h_t: the hallucinated image, which is the image generated from the scratch. It can be expressed as $h_t = H(G(x)_{t-L}^{t-1}, x_{t-L}^{t})$

m_t: the occlusion mask, which can be expressed as $m_t = M(G(x)_{t-L}^{t-1}, x_{t-L}^{t})$, with continuous values between 0 and 1. M is the mask prediction function.

w_{t-1}: optical flow from $G(x)_{t-1}$ to $G(x)_t$, which can be expressed as $w_{t-1} = W(G(x)_{t-L}^{t-1}, x_{t-L}^{t})$. W is the optical flow prediction function.

Therefore, the first part $m_t h(t)$ is used to generate hallucinate new pixels. The second part $(1 - m_t)w_{t-1}(G(x)_{t-1})$ corresponds to the pixels that warped from the previous frames. Here, our occlusion mask M is soft, which can better handle the zoom in and zoom out scenario. To be specific, if we only warp the previous frames, the object will become blurrier when the source person is moving closer to the camera. We need to synthesize new details in order to increase the resolution of the person. Therefore, we can add details by gradually blending the warped pixels and the newly synthesized pixels through using a soft mask.

5.3. Image to Image Translation

In image to image translating stage, we design a system to learn the mapping from the normalized pose stick figures to images of the source person with adversarial training.

Now we detail our training system as shown in the Training setup of Figure 1. We use pose detector P to obtain a corresponding pose stick figure $x = P(y)$ given frame y from the original source video. During training, we use corresponding (x, y) pairs to learn a mapping G which synthesizes images of the source person given pose stick x.

Through adversarial training with discriminator D and a perceptual reconstruction loss distance using a pretrained VGGNet, we optimize the generated output $G(x)$ to resemble the ground truth target subject frame y. Discriminator D attempts to distinguish between real image pairs (i.e. (pose stick figure x, ground truth image y)) and fake image pairs (i.e. (pose stick figure x, model output G(x)).

Our transfer process is also shown in Figure 1. Similarly to training, pose detector P extracts pose information from target frame y yielding pose stick figure x. Then we adopt pose normalization to transform the targets original pose x to be more consistent with the poses in the source video x. We then pass the normalized pose stick figure x into our trained model G to obtain an image $G(x)$ of our target person which corresponds with the original image of the source y.

5.4. Objective Function

We train the sequential video synthesis function Q by solving

$$\min_{F} \max_{D_I} L_I(Q, D_I) + \max_{D_V} L_V(Q, D_V)) + \lambda_W L_W(Q) \tag{10}$$

where L_I is the loss on images defined by the image discriminator D_I, L_V is the loss on video defined by the video discriminator D_V, and $L_W(Q)$ is the flow estimation loss.

The image-conditional GAN loss L_I is

$$\begin{aligned} L_I = {} & E_{\phi I(y_1^T, x_1^T)}[\log D_I(y_i, x_i)] \\ & + E_{\phi I(G(x)_1^T, x_1^T)}[\log(1 - D_I(G(x)_i, x_i))] \end{aligned} \tag{11}$$

Similarly, the video-conditional GAN loss L_V is

$$\begin{aligned} L_V = {} & E_{\phi V(w_1^{T-1}, y_1^T, x_1^T)}[\log D_V(y_{i-K}^{i-1}, w_{i-K}^{i-2})] \\ & + E_{\phi V(w_1^{T-1}, G(x)_1^T, x_1^T)}[\log(1 - D_V(G(x)_{i-K}^{i-1}, w_{i-K}^{i-2}))] \end{aligned} \tag{12}$$

Recall that we synthesize the video $G(x)_1^T$ by applying Q recursively. The flow loss L_W consists of two parts. One is the warping loss during the flow warps from the previous frame to the next frame, the other is the endpoint error between the ground truth and the estimated flow.

5.5. Feature Matching Loss

In equation (12), in addition to the loss terms, we use the discriminator feature matching loss[21] and VGG feature matching loss[22] since they can improve the convergence speed and training stability.

Through adversarial training with discriminator D and a perceptual reconstruction loss dist using a pretrained VGGNet, we optimize the generated output $G(x)$ to resemble the ground truth target subject frame x. For VGG feature matching loss, we use

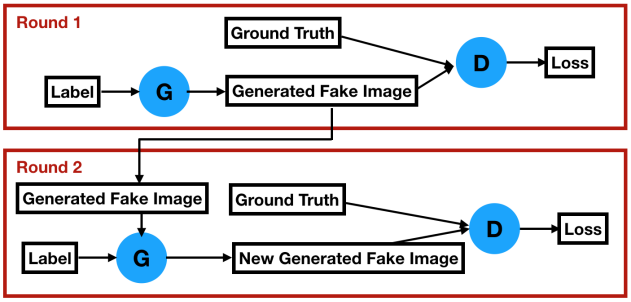

Figure 4. Two stages coarse to fine training

Figure 5. Pose generated for the face and the ground truth face image. The Densepose extracted pose provided the label for the left and rigth face.

the VGG network as a feature extractor and minimize L1 losses between the extracted features from the real and the generated images. In particular, we add $\sum_i \frac{1}{P_i}[||\Psi^{(i)}(x) - \Psi^{(i)}(G(x))||_1]$ to our objective, where $\Psi^{(i)}$ denotes the i-th layer with P_i elements of the VGG network.

6. Network Architecture

The neural network consists of two main parts, i.e. Generator and Discriminator. The generator tries to produce fake images to fool the discriminator and the discriminator learns to distinguish the real and fake images. In this paper, we adopt an course-to-fine training mechanism to train the generate images. In the first round, we set the densepose labels and real image as input and send them to the generative model to generate fake images. The fake image should have similar appearance as the real images. Since the fake images contain more detailed information about the person and background than openpose labels, in the second round, we treat both the openpose labels and first-round generated images as input and retrain the whole model. The stage-round results should much better than the first-round results.

6.1. Generator

We divide our training process in two stages. In the first stage, the network takes in a number of extracted poses x_{t-L}^{t} and generated frames $G(x)_{t-L}^{t-1}$ in last L steps as input. The extracted poses are concatenated together and undergo several residual blocks to form intermediate high-level features. We apply the same processing for the previously generated images. Then, these two intermediate layers are added and fed into two separated residual networks to output the intermediate image $\tilde{h}_t$ as well as the flow map $\tilde{w}_t$ and the mask $\tilde{m}_t$. The flow map is used to warp the previous frames, and then combine with the intermediate frame to output the final frame. The final image then used as input to generate next frame and so on. In the second stage, we use the extracted pose, images generated from last L steps and the output of the first stage (generated image) as the new input. Training the model in the similar way as aforementioned method, we can get well-trained generator.

6.2. Discriminator

As using multiple discriminators has been shown beneficial in mitigating the model collapse problem in GAN training [23][24][25]. We use two types of discriminators in our approach, one for images and one for video. The purpose of image discriminator D_I is to ensure that each output frame resembles a real image given the same source image. For D_I, we adopt the multi-scale PatchGAN architecture, it takes both input maps and output images and evaluate multiple feature scales similar to pix2pixHD. This conditional discriminator should output 1 for a "true" pair (y_t, x_t) and 0 for a "fake" pair $(G(x)_t, x_t)$. The purpose of video discriminator D_V is to ensure that consecutive output frames resemble the temporal dynamics of a real video given the same optical flow. It is also a conditional discriminator. While D_I conditions on the source image, D_V conditions on the flow. Let w_{t-K}^{t-2} be $K-1$ optical flow for the K consecutive real images x_{t-K}^{t-1}. This conditional discriminator D_V should output 1 for a "true" pair $(x_{t-K}^{t-1}, w_{t-K}^{t-2})$ and 0 for a "fake" one $(G(x)_{t-K}^{t-1}, w_{t-K}^{t-2})$.

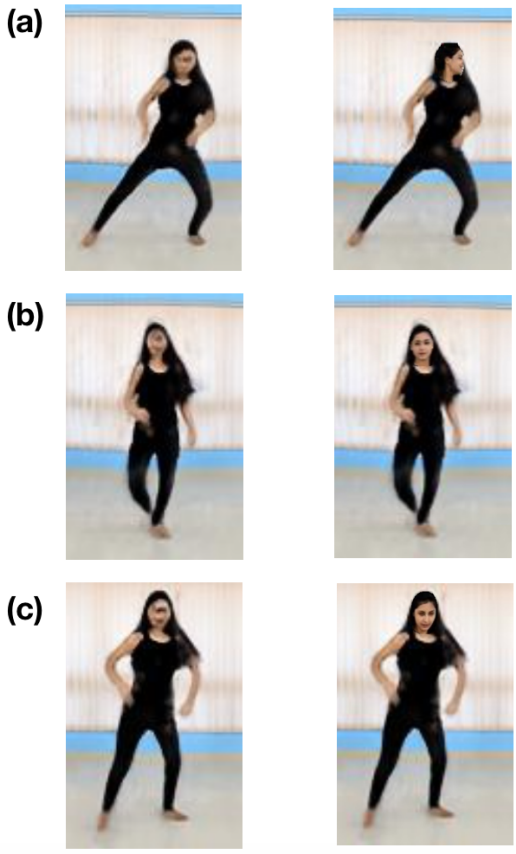

Figure 6. Generated image without face transferring (left) and with face transferring (right). Transferred face from ground truth gives much more details.

7. Face Transferring

As most details are lost in the face in the generated video, we add another trick to directly transfer the face image from the ground truth video in the test set to the generated video. The intuition of this transferring is that with different poses, the details on the face is not changed. The only difference on face between different poses is the direction of the face. So if we can search among the ground truth images for the face with most similar direction, we can directly transfer the ground truth face to substitute the generated face. The similarity defined for two faces is:

$$r = \frac{S_{left}}{S_{right}}$$

which is the ratio of the area between the left and right faces. And this ratio can define the direction of the face, as can be seen in Figure 5.

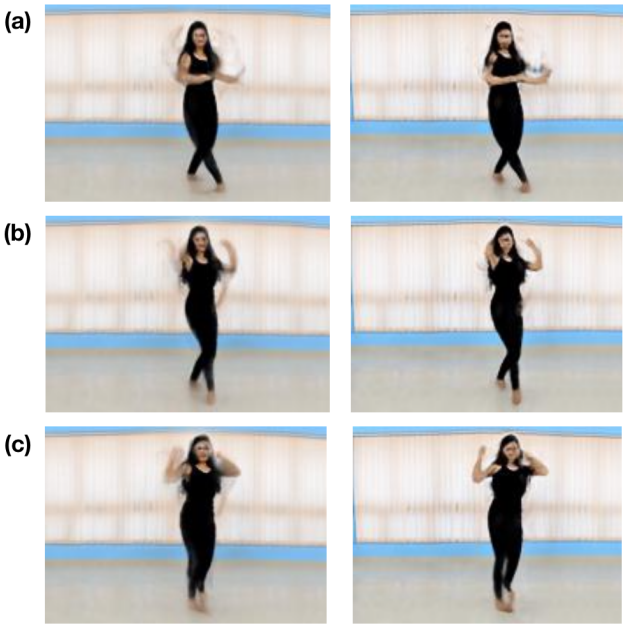

Figure 7. (a)-(c) shows three different examples of generated figures. On the left side are the images generated by the model after the first-round training. On the right side are the images generated by the model after the 2 rounds training.

Some examples of the transferred image are shwon in Figure 6

8. Experiments

8.1. Course-to-fine Training

We jointly train the generator and discriminator. During evaluation, we extract the densepose of the source person and take the densepose as input to generate fake images. The fake images show the target person but the pose is extracted from the target person.

In our first-round training, we take the densepose labels and ground images of the source person as input to generate fake images. In the second-round training, we send the generated fake images, densepose labels and ground images of the source person to the model to train the whole network.

8.1.1 Qualitative Comparison

We compare the results generated by first-round training and second-round training. Some examples are shown in Figure 8. The left column is the pose extracted from the true image, the middle column is the

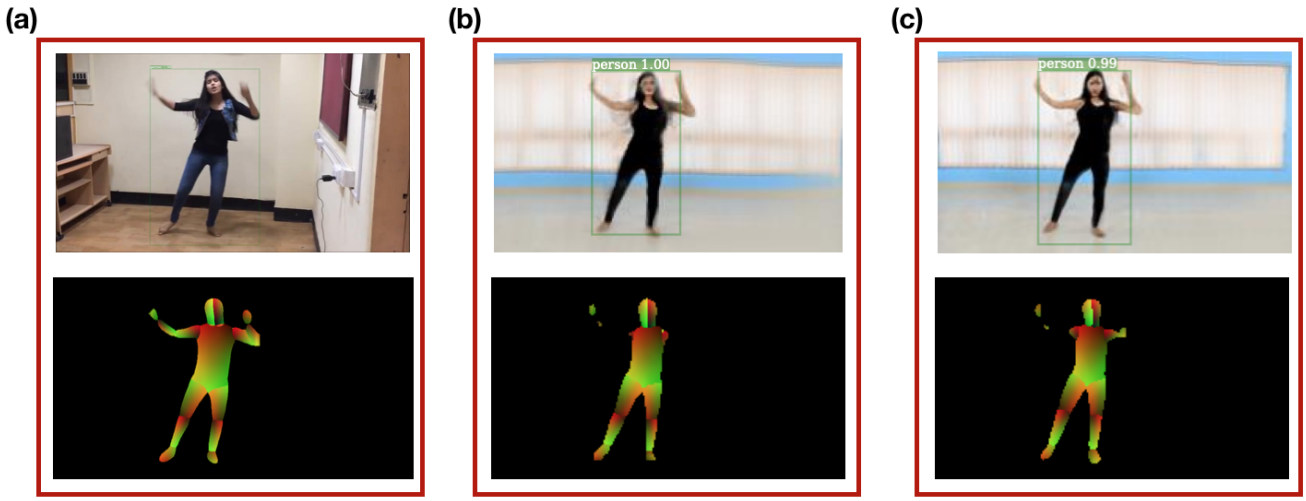

Figure 8. Evaluation based on the IoU of the generated pose. (a) shows the source video and the extracted pose. (b) and (c) show the generated figure by model after round 1 and round 2 respectively and below are there extracted pose. We use the IoU of the generated pose to evaluate the generated images.

pose of the first-round results and the right column is the second-round pose results. In the first-round pose result, the arms, especially the right arm, of the girl are not detected. But we can see the second-round results contains both arms which is closer to the real pose. The results show the boundaries are clearer and the poses are more accurate in the second-round results.

8.1.2 Quantitative Comparison

To quantitative evaluate our proposed course-to-fine mechanism, we propose a new evaluation metric to measure the distance between the generated poses and ground truth poses.

We first extract the densepose labels x_0 for the source person. After first-round training, we transfer the pose from the source person to the target person using our generative model. For the generated target person who has the same pose as the source person, we use densepose again to extract the pose labels x_1. We compute the IoU (Intersection over Union) between the original densepose and the generated pose.

$$IoU = \frac{x_0 \cap x_1}{x_0 \cup x_1} \tag{13}$$

We use the same strategy to compute the IoU between the original densepose and densepose result generated in the second-round. The larger the IoU is, the closer between the generated pose and ground truth pose. In our experiment, we found that the IoU of the second-round is 10% higher than the first-round which verify the efficiency of our proposed course-to-fine training method.

9. Conclusion

In this project, we proposed a course-to-fine training methods for human pose transferring. We train a generative model to generate images according to the input human poses. To accelerate the training process of the generative model, we use low-resolution images as input and output to train the model. During test stage, we extract poses from source person and apply this pose to the target person. We quantitatively and qualitatively compare our results generated by the first-round and second-round training. The second-round training improve the accuracy by 10% which shows the efficiency of our proposed course-to-fine method. We refine the result by proposing a face transfer method from the ground truth images to the generated images. Some generated results are shown in this paper.

References

[1] MehdiMirza BingXu DavidWarde-Farley Sherjil Ozair Aaron Courville Yoshua Bengio Ian Goodfellow, Jean Pouget Abadie. Generative adversarial nets. In *Advances in neural information processing systems*, 2014.

[2] Rob Fergus et al. Emily L Denton, Soumith Chintala. Deep generative image models using laplacian pyramid of adversarial networks. In *Advances in neural information processing systems.*, 2015.

[3] Ferenc Huszr Jose Caballero Andrew Cunningham Alejandro Acosta Andrew Aitken Alykhan Tejani Johannes Totz Zehan Wang et al. Christian Ledig, Lucas Theis. Photo-realistic single image super-resolution using a generative adversarial networ. In *ArXiv e-prints 609.04802*, 2016.

[4] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. In *ArXiv e-prints 411.1784*, 2014.

[5] Michele Covell Christoph Bregler and Malcolm Slaney. Video rewrite: Driving visual speech with audio. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 353-360, 1997.

[6] Greg Mori Alexei A. Efros, Alexander C. Berg and Jitendra Malik. Recognizing action at a distance. In *IEEE International Conference on Computer Vision*, 2003.

[7] Duygu Ceylan Ruben Villegas, Jimei Yang and Honglak Lee. Neural kinematic networks for unsupervised motion retargetting. In *arXiv preprint arXiv:1804.05653*, 2018.

[8] C. Chan, S. Ginosar, T. Zhou, and A. A. Efros. Everybody Dance Now. *ArXiv e-prints*, August 2018.

[9] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *arXiv preprint arXiv:1409.1556*, 2014.

[10] E. Mansimov N. Srivastava and R. Salakhudino. Unsupervised learning of video representa- tions using lstms. In *International Conference on Machine Learning (ICML)*, 2015.

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

[12] A. Szlam E. Denton, S. Chintala and R. Fergus. Deep generative image models using a laplacian pyramid of adversarial networks. In *Advances in Neural Information Processing*, 2015.

[13] K. Bouman T. Xue, J. Wu and B. Freeman. Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

[14] A. Shamir T. Chen, J.-Y. Zhu and S.-M. Hu. Motion-aware gradient domain video composition. In *IEEE Trans. Image Processing, 22(7):25322544*, 2013.

[15] D. Ha and J. Schmidhuber. World models. In *arXiv preprint arXiv:1803.10122*, 2018.

[16] O.Poursaeed-J.E.Hopcroft andS.J.Belongie X.Huang, Y.Li. Stacked generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[17] Y. Caspi E. Shechtman and M. Iranie. Space-time super-resolution. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 27(4):531545*, 2005.

[18] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Video-to-video synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[19] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[20] A. Alahi A. Gupta, J. Johnson and L. Fei-Fei. Characterizing and improving stability in neural style transfer. In *arXiv preprint arXiv:1705.02092*, 2017.

[21] T. Breuel M.-Y. Liu and J. Kautz. Unsupervised image-to-image translation networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

[22] I. Goodfellow C. Finn and S. Levine. Unsupervised learning for physical interaction through video prediction. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

[23] M. Arjovsky V. Dumoulin I. Gulrajani, F. Ahmed and A. C. Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

[24] C. Vondrick and A. Torralba. Generating the future with adversarial transformers. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[25] P. Dollr Z. Tu S. Xie, R. Girshick and K. He. Aggregated residual transformations for deep neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.