SCENE-PRESERVING PERSON APPEARANCE TRANSFER

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

FAHRIYE ÖZGE ÜNEL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

FEBRUARY 2021

ABSTRACT

SCENE-PRESERVING PERSON APPEARANCE TRANSFER

Ünel, Fahriye Özge
M.S., Department of Computer Engineering
Supervisor: Assist. Prof. Dr. Ramazan Gökberk Cinbiş

February 2021, 89 pages

Recent developments in deep learning and deep generative models have enabled numerous new applications in the area of image and video editing. One such emerging topic is the editing of pose and appearance of the person across images. Several recent works have introduced methods for transforming the pose of a single person within the same scene. In addition, there are works that aim to model the appearance, scene and pose through separate representations, and produce new images with arbitrary appearances, backgrounds or poses through these representations. Similarly, some works model the appearance of a particular person through his/her video to learn to generate the person in arbitrary poses. In this thesis, we tackle a comprehensive version of these problems by aiming to learn a generative model that can replace the person in an image with another person from a single image, both with possibly different poses and different backgrounds. More specifically, we aim to learn a generative model that takes a single image of an *actor* with an arbitrary pose and a *stuntman* that provides the pose and scene information and yield a new image that contains the background scene and pose of the stuntman and the appearance of the actor. We aim to obtain a realistic final image, which requires properly re-generating the actor appearance in the pose of the stuntman and synthesizing the missing background

and foreground pixel information due to pose and physical characteristic differences. For this purpose, we propose an end-to-end framework that is trained to maximize the prediction quality of pixel-wise foreground and background details via masked reconstruction loss terms and realism of the output image via an adversarial trained discriminator network. We also introduce a new benchmark by adapting the video segmentation datasets YouTube VOS and Davis for the proposed task. We experimentally investigate and evaluate our approach on the proposed benchmark dataset.

Keywords: human pose transfer, person pose transfer, pose guided person image generation, background preserve human generation, person image generation, generative models, generative adversarial networks

ÖZ

SAHNEYİ KORUYAN KİŞİ GÖRÜNÜM AKTARIMI

Ünel, Fahriye Özge
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi: Dr. Öğr. Üyesi. Ramazan Gökberk Cinbiş

Şubat 2021 , 89 sayfa

Derin öğrenme ve derin üretken modellerdeki son gelişmeler, görüntü ve video düzenleme alanında çok sayıda yeni uygulamayı mümkün kılmıştır. Kişinin poz ve görünüşünün görüntüler arasında düzenlenmesi bu tür konulardan bir tanesidir. Son zamanlarda yapılan birkaç çalışma, aynı sahnede tek bir kişinin pozunu dönüştürmek için yöntemler ortaya koymuştur. Ayrıca görünüşü, sahneyi ve pozu ayrı temsiller üzerinden modellemeyi ve bu temsiller üzerinden keyfi görünümler, arka planlar veya pozlarla yeni imajlar üretmeyi amaçlayan çalışmalar vardır. Benzer şekilde, bazı çalışmalar, bir kişinin videosunu kullanarak kişiyi rastgele pozlarda oluşturmayı öğrenmek için belirli bir kişinin görünümünü modellemektedir. Bu tezde, bir görüntüdeki kişiyi hem muhtemelen farklı pozlar hem de farklı arka planlara sahip tek bir görüntüden başka bir kişiyle değiştirebilecek üretken bir model öğrenmeyi hedefleyerek bu sorunların kapsamlı bir versiyonunu ele alıyoruz. Daha spesifik olarak, rastgele bir poza sahip bir aktörün tek bir görüntüsünü ve poz ve sahne bilgilerini sağlayan dublörün görüntüsünü kullanarak aktörün kendi görünümünde, dublörün pozunda ve arka planında görünüşünü içeren yeni bir görüntüsünü oluşturan üretken bir model öğrenmeyi hedefliyoruz. Dublör pozundaki aktörün görünümünün düzgün bir şekilde

yeniden oluşturulmasını ve poz ve fiziksel özellik farklılıklarından kaynaklanan eksik arka plan ve ön plan piksel bilgilerinin sentezlenmesini gerektiren gerçekçi bir son görüntü elde etmeyi hedefliyoruz. Bu amaçla, maskelenmiş rekonstrüksiyon kaybı aracılığıyla piksel bazlı ön plan ve arka plan ayrıntılarının tahmin kalitesini maksimize etmek ve rakip eğitimli bir ayırt edici ağ aracılığıyla üretilen görüntünün gerçekliğini en üst düzeye çıkarmak için eğitilmiş uçtan uca bir çerçeve öneriyoruz. Ayrıca, video segmentasyon veri seti olan YouTube VOS ve Davis'i önerilen görev için uyarlayarak yeni bir karşılaştırma ölçütü sunuyoruz. Yaklaşımımızı önerilen kıyaslama veri setlerinede deneysel olarak inceliyor ve değerlendiriyoruz.

Anahtar Kelimeler: insan poz transferi, poz tabanlı insan üretimi, arka plan koruyarak insan üretimi, insan resmi üretimi, üretici modeller, çekişmeli üretici ağlar

ACKNOWLEDGMENTS

First of all, I would like to deeply thank my supervisor Asst. Prof. Dr. Ramazan Gökberk Cinbiş for his efforts in supporting me with his generous help and guidance. Writing this thesis would have been impossible without his support, knowledge and experience.

In addition, I am grateful to my thesis committee members Asst. Prof. Dr. Emre Akbaş and Asst. Prof. Dr. Selim Aksoy for reviewing this thesis and their feedback and contributions.

I am thankful to all of my friends for their understanding throughout this process.

Finally, I am grateful to my family and Kemal for their kind respect and encouragement and being my best supporter.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

Image generation is a difficult problem due to the need to produce content that has never been seen before. Early approaches are based on computer graphics methods, and among these methods, transform-based methods [6, 7, 8] learn the transformation matrix of objects from different points of view, while geometric methods [9, 10] use 3D information to reconstruct the object. Recent advances in deep learning and deep generative models have shown that it is possible to construct models that yield realistic synthetic images based on simple statistical learning principles, which makes a variety of new applications possible.

Person image generation is one of the most challenging and most practically valuable topics in the spectrum of image generation problems. This problem is of great importance in a variety of applications such as image editing, face editing, generation of the invisible parts of objects, surveillance data augmentation , and video forecasting [11, 12]. Transforming a person into a given pose, which is commonly referred to as *pose guided person generation* or *human pose transfer* [13, 1, 4, 14], is one of these applications.

In this section, we make an introduction to the problem of human pose transfer and mention the reasons why there have been many studies on this problem recently. We then briefly discuss well-known approaches on this topic and the main challenges. After defining the limitations of existing pose transfer methods, we provide a brief summary of the approach proposed in this thesis, and clarify our contributions. We conclude the chapter with an outline of the thesis.

1.1 Human Pose Transfer

The main purpose of human pose transfer in images is defined as transforming the pose of a particular person's image into the desired pose while preserving the appearance and attributes such as the person's hair, body structure, and clothes. The deep neural networks that aim to achieve this goal successfully have three different inputs that need to be known within the scope of pose transfer. The first input is the source image that gives the information about the person's appearance. The second input is the target person image from which pose information will be obtained. While generating a person given as a source into a target person pose, heat map or stickman, which is a representation of the pose, is obtained from the image of the target person and is given to the network. These representations provide implicitly the pose information through an image. Furthermore, the effect of the appearance information of the target person is restricted by including the geometry of a human skeleton only.

To transfer the human pose of a particular image to another, deep neural network models that perform pose transfer extract pose information from the stickman or heatmap image and apply this extracted information to transform the source image without changing the human appearance details of that person's image. Heatmap is preferred in many studies in the literature because this pose representation keeps each joint information in a separate channel and thus provides a 3D representation.

Using the source person image and the pose information of the target person, Ma et al. [1] and Zhao et al. [15], two of the pioneers of contemporary developments in the field, synthesize the image of the source person in the expected pose, without losing the person's appearance and attributes. In the follow up of these works, there has been a rapidly growing interest in this topic, *e.g.* [16, 17, 4, 18, 19, 20, 21, 14].

The reason for the extensive interest in the field is that there are many application areas such as fashion design, computer graphics based manipulations [11], person re-identification (Re-ID) [22, 23], human pose estimation systems [24, 25]. Deep neural networks can model the distribution and properties of the data set given during the training phase and make predictions on the given data using the information gathered from the training phase at test time. For this reason, the data set used in model training

should be as rich as possible in order to provide a sufficiently comprehensive coverage of a wide range of poses. This coverage directly affects the results in terms of sharpness, details and realism for person/human pose transfer, which fundamentally alters the value of human pose transfer in producing synthetic training examples for person Re-ID and human pose estimation.

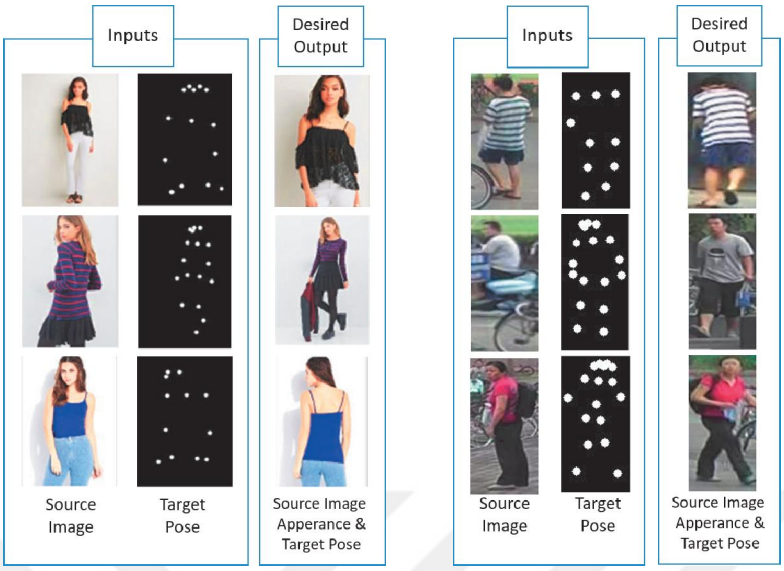
Contemporary deep generative models, such as *Variational Auto-encoders* (VAEs) [26] and Generative Adversarial Networks (GANs) [27], provide powerful tools for realistic image generation. Generative models aim to learn the true data distribution underlying the training set and novel generative data points from the same distribution. Several studies, such as [28, 29, 30], utilize the ability of these networks to create sharp images that help synthesizing photos more realistically.

A variety of deep generative model based pose transfer approaches have recently been proposed in the literature. To give an overview of these recent developments, we categorize these approaches under four different topics based on the specific problem that they are tackling. Below, we list and briefly explain these four categories.

1. **Source Appearance Aware Human Pose Transfer (SAAHPT).**

   When transforming a person into a different person's pose, approaches in this group only aim to model that person's appearance. Studies that adopt this category do not consider the scene information of the source or the target person, so they only get the image of the source person and the pose information of the target person as an input to the neural network and produce the image of the source person in the target pose without deforming the appearance of the source. There are many studies such as [13, 1, 15, 31] on this subject.

   Figure 1.1 shows the inputs and desired outputs of SAAHPT separately in Market [32] and Fashion data [33]. In this figure, the first two columns are an input pair consisting of the source person image and the heatmap of the desired pose. The third column shows the desired output, which depends on both the appearance information in the source image and the target pose.



(a) DeepFashion (b) Market-1501

Figure 1.1: Showing an example of human pose transfer with the SAAHPT approach. Figure taken from [1].

2. **Source Background Aware SAAHPT (SBA-SAAHPT).** In addition to the first approach, this method preserves the background of the source person. Its inputs and outputs are the same as the first method. [2] is one of the studies on this topic. Figure 1.2 shows the inputs and outputs of SBA-SAAHPT in Market dataset.

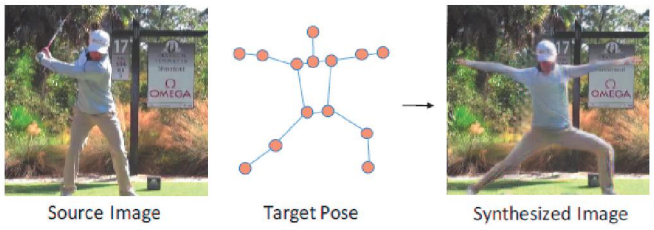


Figure 1.2: Showing an example of human pose transfer with the SBA-SAAHPT approach. Figure taken from [2].

3. **Target Background Aware SAAHPT (TBA-SAAHPT).** In addition to the first approach, this method protects the background of the target person. This

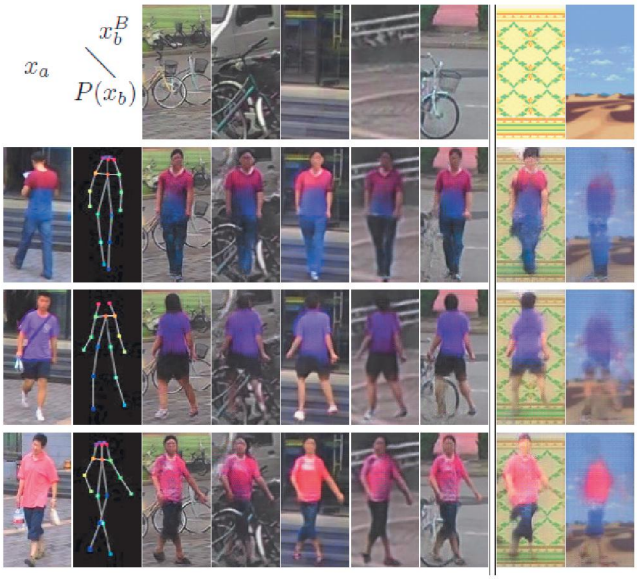


Figure 1.3: Showing an example of human pose transfer with the TBA-SAAHPT approach. Figure taken from [3].

approach group use the image of the target person as well as the inputs used in the first approach. Some studies on this topic are [3, 34, 35] . Figure 1.3 shows the inputs and outputs of TBA-SAAHPT in Market dataset.

4. **Arbitrary Human Pose Transfer (AHPT).** It is an approach designed to produce images in different scenes, appearance and poses by changing with one of them after learning the foreground, background and pose information over the data set. This method is not used to replace a person or scene with a new one, but to manipulate a person to be created in a random scene or view, or to provide interpolation between two different people, scenes or poses. [4] is one of the first studies on this topic. Figure 1.4 shows the results of Mae et al. [4] study on the Market dataset.

Main challenges. In the human pose transfer task, the generated person is expected to comply with the conditioned pose structure as well as preserving the appearance details of the source person. This task is challenging, as it requires image translations with non-trivial movement variations, and even challenging changes with the only semantic similarity between images. While transferring the pose, the generation net-

Figure 1.4: Figure shows the results on the Market dataset in different poses, appearance and scenes. Figure taken from [4].

work has to deal with misalignments in target-to-source pixel conversion due to the pose differences, as well as the problems observed during human generation, such as the high dimensionality of the pose, the complexity of human skeletal structure, cluttered background, local details, global structures, self-occlusion, and occlusion by the other objects. Many published studies [13, 3, 18, 34, 36, 37, 38] attack one or more of these issues.

1.2 Our Proposed Approach to Human Pose Transfer

Problem definition. Most of the methods in the literature focus solely on creating a person in a particular pose. Since these methods aim to generate novel poses with the usage of pose information only, they can not perform well on the scene remaining in the background. Because the background information is not given as input to the

network, this situation causes a blurry background that fits the distribution of the data set and damages the quality of the entire image. Unlike many studies in the literature, our approach focuses on creating a human image by changing the appearance, pose, and background.

Some studies in the literature have worked on the realistic generation of the scene as well as the human appearance. *RHPT*-based approaches manipulate the image to synthesize a person's image with arbitrary but meaningful backgrounds while producing in a different pose with a similar appearance. These kinds of studies aim to produce images of people with different but realistic backgrounds from random distribution instead of producing a person in the desired background. Moreover, *SBA-SAAHPT*-based methods produce an image of the source person while preserving the background of that image and fill the pixel gaps caused by the change of the person's pose in accordance with the scene. Contrary to these approaches, we produce a person's image in the desired pose and scene. We differ from *TBA-SAAHPT*-based approaches, which have a similar purpose to our study, with the methods and inputs we use. Details are explained in Section 2.3.

Our approach. In this thesis, we address the problem of generating a person image using the appearance of a person in a source image, conditioned on a given target person pose and background information. This situation can be thought of as replacing a stunt scene with a famous actor. Specifically, our aim is to synthesize a human image by satisfying two conditions: The first one is the preserve the appearance information of a given person under a different pose. The second condition involves producing the source image according to the desired human pose and background. The task attacked by our proposed network in this thesis is to overcome the problems of creating the background while maintaining the appearance of the source image (eg. texture, clothing, body structure) with the target pose.

By using the image of the source and the target as well as the target pose information, our network both transforms the person's appearance information in the source image into the pose of the target person and emplaces the person in the source image on the scene of the person in the target image. Therefore, unlike previous studies, it is additionally given to the deep neural network of the target person as input. The system

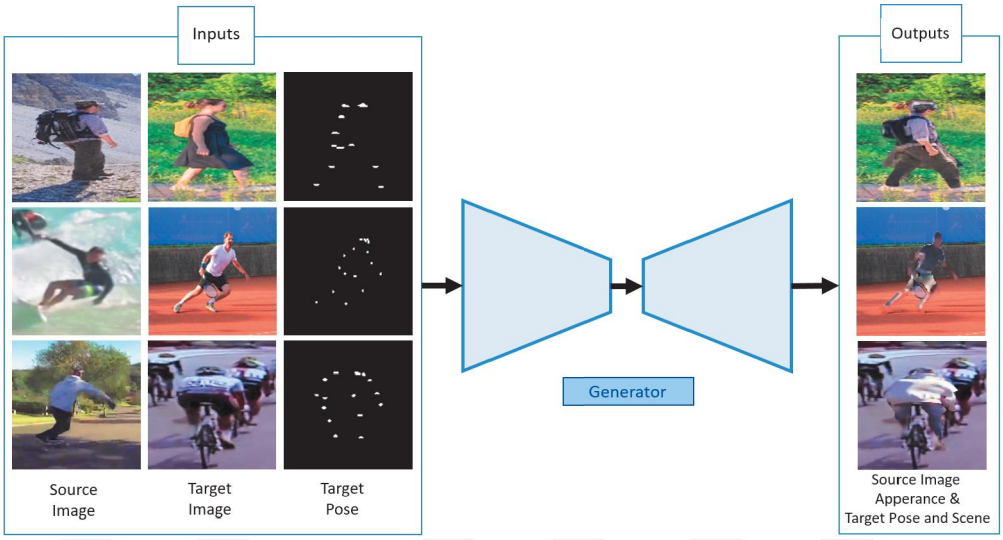


Figure 1.5: Showing an our proposed approach for human pose transfer. The first column represents the source person's image, the second column represents the target person's image, and the third column represents the target person's pose heatmap. Our approach uses the pictures in the first three columns to produce the image shown in column 4, which has the appearance of the source person and the pose and background of the target person.

shown in Figure 1.5 uses the source image and the image of the target as input, as well as pose information extracted from the target image. Our experiments show that our method can generate realistic human images but also produce the human pose and background as we want.

1.3 Contributions

Major contributions of this thesis are summarized below.

- We propose a person's image-generating framework to synthesize a new person image of that person in the novel target pose and, unlike other studies, the background information according to the image given to the target when an image of a person and a target person is given.
- We make an extensive analysis on the human image generation of different

loss types (hinge, bce, wgan-gp), normalization methods (batch norm, instance norm, spectral norm), different architectural designs, and hyperparameter selections.

- We propose a novel joint mask loss to encourage the model to separate the human body view from the source image and the background view from the target image. And we only used it during the training phase.
- We present a comprehensive qualitative and quantitative experimental evaluation of the approach, including ablative studies to verify the effectiveness of the approaches we propose.
- We also introduce a new benchmark by adapting the video segmentation datasets YouTube VOS and Davis to create a person image in the pose and background of the desired person.

1.4 Organization of the Thesis

The organization of the thesis is as follows. In Chapter 2, we briefly talk about the previous work about synthesizing an image of a person conditioned on both pose and appearance information. In Chapter 3, we explain our approach with technical details. In Chapter 4, we present out experimental results and finally in Chapter 5, we conclude the thesis with a discussion of possible future work directions.

CHAPTER 2

RELATED WORK

In this chapter, we first present an overview of the generative models used in the Image-to-Image (I2I) generation problem, and then give an extensive explanation of GAN and U-NET models since we later use these frameworks in the construction of our guided I2I generation network. Finally, we present a comprehensive summary of the studies conducted on human pose transfer.

2.1 Generative Model

In this section, we make an introduction to notable approaches for generative modeling and briefly talk about the types and usage areas. A Generative Model defines distributions over a set of variables using unsupervised learning methods. These models aim to learn the real data distribution underlying the dataset during the training phase, then use this learned distribution to produce new data points. These models are extensively utilized in several different applications such as movie making, face editing, and super-resolution, text to the image. There are three effective methods for this task: (Variational) Autoencoders (AEs/VAEs) [26], Autoregressive models (e.g., PixelRNN [39], PixelCNN [40]), and Generative Adversarial Networks (GANs) [27]. These frameworks allow us to create data points similar to real data points.

Each mainstream approach is summarized in the following sub-sections providing the drawbacks and give certain details about the recent advances in image generation topic.

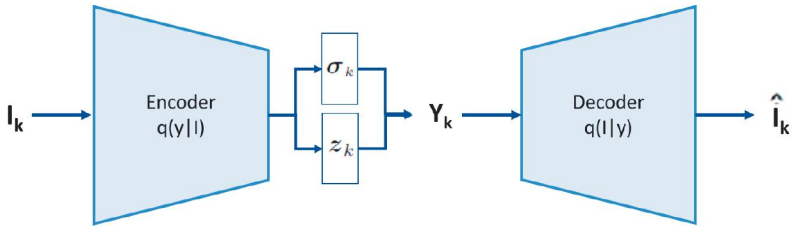


Figure 2.1: Structure of original Variational encoder that uses the spatial image domain.

2.1.1 Review of (Variational) Auto Encoder

Recently, auto-encoders are extensively utilized in several different machine learning applications from image denoising [41] to feature extraction [42], thanks to their ability to represent high dimensional inputs with significantly lower-dimensional components. After the initial success of the vanilla auto-encoders, variational auto-encoders are introduced to extract latent information in the input data using probabilistic perspective. Unlike the vanilla auto-encoders, where a high dimensional input is summarized by a vector with much smaller dimensions, in variational encoders, one tries to find the latent components or distribution, which generates the input data. In other words, variational encoders try to find the underlying hidden distribution that generates the input data.

For image domain application, the auto-encoder first processes the image which is represented by I_k in order to extract the intrinsic information as shown in Figure 2.1. Then, the encoded information is decoded in order to reconstruct I_k. The reconstructed images are represented as $\hat{\mathrm{I}}_k$. An encoder and decoder structure is designed so that the I_k is close to the $\hat{\mathrm{I}}_k$ in a certain error sense. *The Mean Square Error (MSE)* or *Mean Absolute Error (MAE)* is typically used to measure the error between original and reconstructed errors. Note that y_k represents the encoded image where the dimension of y_k is much smaller than I_k. y_k is calculated by the Equation (2.1):

$$\vec{y}_k = \vec{z}_k + e^{\vec{\sigma}_k}\epsilon; \epsilon \sim \mathcal{N}(0,\ 1) \tag{2.1}$$

where ϵ is a normal distributed random variable with zero mean and standard deviation 1, $\vec{z}_k$ represents the mean and $\vec{\sigma}_k$ represents the variance of this distribution.

Although these models allow us to formalize this problem in the framework of probabilistic graphical models, the results tend to be blurry mainly as a result of MSE loss.

2.1.2 Autoregressive Models

There are two well-known contemporary autoregressive models in the domain of generative image modeling. The first one is PixelRNN [39], which learns through the network that models the conditional distribution of each pixel given the previous ones. And this model runs both horizontally and vertically on the image. PixelRNNs have a very basic and stable training process and do not need a training phase to learn the distribution of all data. However, generation is typically slow since each state needs to be computed sequentially. Another autoregressive model is PixelCNN [40], which models dependency on previous pixels using a convolutional model.

2.1.3 Review of Generative Adverserial Networks (GAN)

Generative Adversarial Networks (GAN) [27] are deep neural networks that consist of two sub-networks, generator, and discriminator. These two networks are trained through an optimization routine in the form of a minimax game, where one network (generator) tries to generate realistic data and the other network (discriminator) tries to discriminate whether some input data is real or fake. As a result of this game, the model learns to create data similar to training data by learning the underlying data distribution.

The architecture of simple GAN is given in Figure 2.2. The generator network (G) gets a random number and produces a fake image. Both this fake image and an image taken from the real dataset is fed to the discriminator network (D). The discriminator takes these images and treats the situation as a binary classification problem. Therefore, the discriminator returns probabilities that are a number between *0 and 1*, where *1* represents a real image and *0* represents a fake image. The discriminator network is the decision-making mechanism and is responsible for updating procedures for both itself and the generator network. To achieve this goal, this network makes two dif-

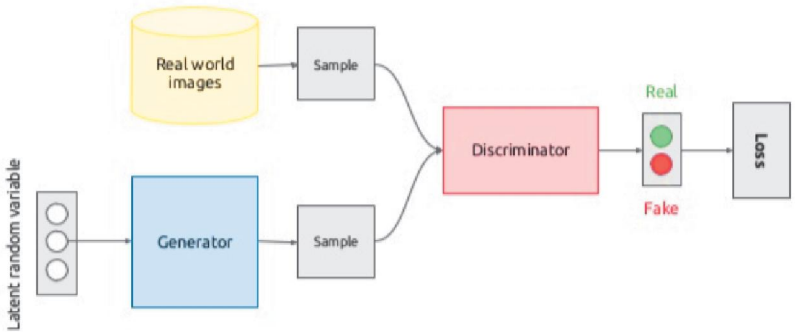


Figure 2.2: Structure of simple Generative Adversarial Network (GAN) which consists of two networks known as generator and discriminator.

ferent updates. One of them is to update its own network, and for this purpose, the D network classifies the data coming from the generator as false, while classifying the real image correctly, providing a better separation of the discriminator network. The other is to update the generator network. The discriminator correctly classifies the image from the G network, encouraging the generator to produce images with appearances similar to real images.

The original GAN loss function used for optimizing the GAN networks is as follows::

$$min_G max_D \mathbb{E}_{x \sim p_r} log[D(x)] + E_{z \sim p_z} log[1 - D(G(z))] \tag{2.2}$$

In this equation, the generator and discriminator networks are represented as G and D, respectively. And, p_r indicates the distribution of the actual data, while p_z indicates the distribution of the hidden random variable. The purpose of the GAN is to train the generator such that it approximates the actual data distribution.

The work done in this area is divided into three categories: unsupervised (unconditional) and semi-supervised and supervised (conditional) settings.

Unsupervised GANs. Goodfellow et.al [27] used fully-connected (FC) layers for both generator and discriminator networks. This work and variants use FC network to generate images from some simple dataset like MNIST [43], CIFAR-10 [44] and Toronto face dataset . Other non-conditional GANs are LAPGAN [45] and DCGAN [28] which use convolutional layers in discriminator and transposed convo-

lutional layers with stride convolutions in generator network to generate images with larger spatial dimensions. Besides, DCGAN uses the Leaky ReLU activation function that allows small gradient transition for negative values in the discriminator network. The reason for this is to ensure that negative values from the discriminator can update the generator. The loss formula is the same as the equation 2.2 mentioned in the original GAN article. Also, several articles [46, 47, 48] perform image-to-image translation aiming to translate an image from one area to another in an unsupervised manner.

Supervised GANs. Some application areas require synthesis of the images with some conditions such as perspective, pose, label, text, images. Using these conditions to indirectly learn different distributions in GANs reduces the likelihood of *mod collapse*. Mirza et al. [49] propose a Conditional Generative Adversarial Network (CGAN) to achieve attribute (class label) conditioned image generation on the MNIST dataset. The CGAN replaces the equation 2.2 with the additional conditional variable y:

$$min_G max_D E_{x\sim p_r} log[D(x|y)] + E_{z\sim p_z} log[1 - D(G(z|y))] \tag{2.3}$$

In addition, conditional GANs are used in many areas, such as render faces [50], cloths [51, 52], pose [37, 17, 38], style [53] and indoor/outdoor scenes [54, 55].

Semi Supervised GANs. The third category uses unsupervised and supervised learning together. Semi-supervised GAN (SGAN) [56] has labels for a small subset of example. Discriminator of SGAN is multi-headed to classify real data and distinguish real and fake samples respectively. In this work, the results of the SGAN framework in the MNIST dataset demonstrated the success of semi-supervised learning by showing that the model performs better at generating hand writing images than the original GAN does.

GAN stabilization and performance enhancement methods. Although GANs typically generate the sharpest images [49, 28, 57, 58, 59] among generative models, they are more challenging to optimize due to unstable training dynamics. Balancing the discriminator and the generator in training is difficult and usually, the discriminator wins easily that causes failure of convergence. Besides, selecting the correct hyperparameters tends to be very important.

There are various studies about solutions to different GAN problems in the literature due to extensive research of GAN. There are also studies on faster and stable GAN trainings. Wang et al. [60] and Kurach et al. [61] examined existing studies according to different design decisions such as losses, architectures, regularization, and normalization. Among the design decisions used to stabilize GANs are studies on architecture [49, 56, 28, 62, 29, 45, 63, 64, 65, 66, 67, 68, 69, 70], studies on loss functions [30, 71, 72, 73, 74, 75], methods of regularization and normalization [30, 76, 77]. [78, 61, 79, 80] have provided empirical studies.

The causes and consequences of the problems observed in GAN are:

- **Mode collapse:** Normally a trained generator network is expected to produce a wide variety of realistic outputs. However, GAN training strategy, unless engineered very carefully, easily results in degenerate generator models, which produce only a limited variety of samples. This situation is called mode collapse. Different techniques are used to prevent this situation. One of these methods is Wasserstein loss [30], aiming to prevent discriminator (also called separator) from getting stuck at a poor local minimum. By updating the discriminator more than once in response to a single update of the generator, UnrollGAN [73] uses a generator loss function that updates not only the current discriminator's classifications, but also the classification outputs of the future discriminator. In this way, it aims to prevent the local minima problem. However, this process slows down the training process as it requires the use of the discriminator multiple times. As a solution to this situation, TTUR approach has been proposed. TTUR [80] aims to prevent the local minima problem by defining larger update rule for the discriminator network.

- **Diminished gradient:** If the discriminator network is over-confident, it may restrict the learning of the generator network because discriminator does not provide enough information to the generator and causes a reduced gradient problem. Wasserstein loss can be used as a remedy for this situation. Alternatively, spectral normalization [77] technique can be applied.

There are also studies of DCGAN [28], one side label smoothing [81], feature matching [81] to solve problems like training instability and mode collapse.

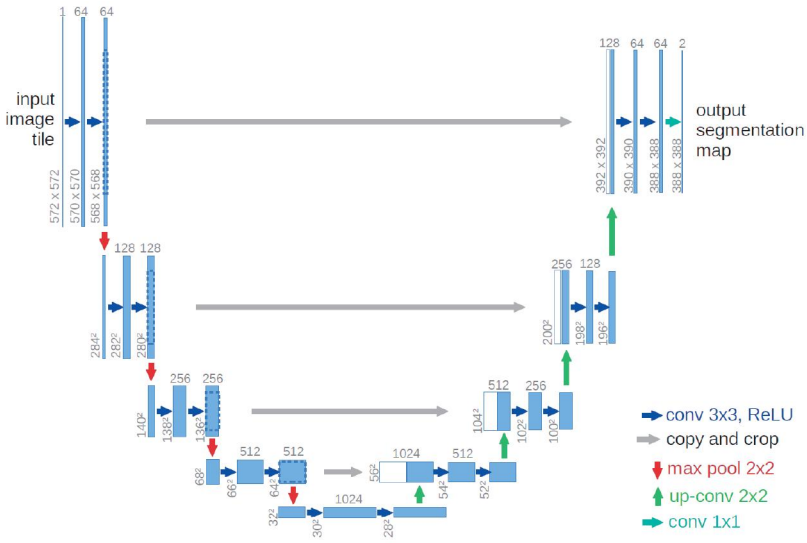


Figure 2.3: Structure of original U-Net taken from [5].

2.2 Review of U-Net

Convolutional auto-encoders are widely used in image manipulation tasks. They are composed of encoder and decoder parts which are responsible for encoding an input image to a lower dimension array and decoding that lower dimension array to output image, respectively. One of the main drawbacks of auto-encoders is that the decoder can generate blurry images. Because high-frequency components of the image are lost due to compression. U-Net [5] architecture solves that problem by defining skip connections that transport low-level features to the deep layer. The image can be restored successfully since low-level features contain high-frequency components. Therefore, U-Net architecture obtains notable success in preserving fine-grained visuals through skip-connections and is the cornerstone of many of the areas from image-to-image translation to segmentation.

The network architecture defined in the original U-Net paper is given in Figure 2.3. The first part of the network is responsible for the down-scaling image with convolution and max-pooling layers. Two convolutions followed by max-pooling are repeated by 5 times. The second part of the network upscales the down-scaled image with the features extracted in the first part.

According to the paper, the U-Net handles the problem in low-level layers by using skip connections. When the input and output images are the same, the high-level features cannot be learned by the network, as it just copies input by using the first layer if the link in the first layer is not removed. Therefore, there is no skip connection in the first layers of networks where memorization problems are experienced.

2.3 Human Pose Transfer

In this section, after providing an overview of person generation and its application areas, we give brief information about one of these applications, pose transfer. Then, we give you an overview of the studies that transfer poses through images and the studies that transfer poses by training through video data in the literature. Then, by classifying the image-based methods according to their approach, we also give an overview of the state-of-the-art studies in these approaches. Finally, we provide works that are similar to our approach in detail and we state our differences to them.

In the literature, various frameworks for person image generation have been proposed. Most studies concentrate on either human pose transfer [15, 1, 4, 19, 13, 82, 2, 21] or virtual try-on [83, 84, 51, 52]. Virtual try-on architectures learn to form a patch clothing to cover the target person, while human pose transfer architectures learn to manipulate pixels to produce a particular person in a different pose without distorting appearance information. For this reason, the pose transfer changes the pose information while preserving the identity of a particular person and physical attributes such as clothing and body structure, but virtual try-on, unlike pose transfer, changes the person's clothes while maintaining the person's pose. Our approach is on pose transfer from these application areas.

Pose transfer methods can also be separated according to different input domains like image-based [85, 35, 86, 87], video-based [88, 89, 37, 38, 36]. Chan et al. [88] learn to represent a person's pose information by using images in different poses from a video. When the pose representation of another person is given to the network after the training phase is over, the person whose pose representation is learned from the video produces according to the information of the pose given. This study has three

disadvantages. The first one is that retraining is necessary when a different person needs to produce images of them in new poses. Another one is that for each person there should be a video of that person in different poses. Finally, the image of the person can not be generated in a different scene, since the training is done through the videos shot in indoor scenes. In this thesis, contrary to these studies, we focus on creating an image of a person on an image using 2D pose information without being dependent on a single person during the training phase.

Novel works on the generation of pose guided person images have made vital progress with GANs [27, 49]. When GAN networks are used alone, images of people in different poses can be obtained through a random input, but although these networks alone can produce a person under the control of a different attribute, they are inadequate to turn a person into a desired person's pose or cloths. Therefore, some studies have combined the capabilities of AE / VAE or Hourglass networks with the capabilities of GAN networks. For the generation of person images, Lassner et al. [31] and Zhao et al. [15] proposed a model that combines GAN and VAE [G2] to create images of a person in different clothes. With the advent of U-Net based architectures, interest in a pose-based person-image generation [1, 3, 11, 20] has gradually increased. This is because, in AE networks, the high-frequency components of the image that are lost due to compression, are causing blurry images, while U-Net architecture uses skip connections similar to the ResNet to prevent this problem.

In the literature, different pose transfer approaches have been proposed by using generative based deep models. We can classify them under 4 different categories as approaches that do not care about the background (SAAHPT), randomly generate the background (RHPT), generate the background from the source person (SBA-SAAHPT), and generate the background from the target person (TBA-SAAHPT). In Table 2.1, we classify the state of the art studies accordingly. Studies that do not care about background are marked as none, while studies that produce a random but meaningful background are marked as arbitrary. Backgrounds produced according to a specific source or target are marked under the relevant column. Some studies may also change the foreground arbitrary, the table is also indicated in such studies. In addition, some studies produce more realistic images by making some improvements on the works in their category. It is given together with the method used in such studies.

	Foreground	Background				
Paper	Arbitrary	None	Arbitrary	Source	Target	Optimization
[1]		✓				
[15]		✓				
[16]	✓	✓				
[4]	✓		✓			
[13]		✓				affine transform
[3]					✓	affine transform
[2]				✓		
[90]		✓				
[14]		✓				normalization
[91]		✓				
[88]		✓				
[34]					✓	affine transform
[92]		✓				
[87]		✓				normalization

Table 2.1: Classification of the latest studies in the literature according to SAAHPT, SBA-SAAHPT, TBA-SAAHPT, and RHPT approaches.

2.3.1 Source Appearance Aware Human Pose Transfer (SAAHPT)

This approach focuses solely on preserving the person's appearance while producing a person in a different pose. Ma et al. [1] and Zhao et al. [15] two of the pioneers of this field have synthesized the image of a person in the expected pose without losing that person's appearance and features, from body structure to clothing. As a work that increases interest in human pose transfer, Ma et al. [1] aim to generate images of a person with different poses in accordance with the same identity. This work provides a two-stage U-Net-based generator network PG2. The first stage is responsible for producing a coarse image by transforming the source person into the pose of the target, and this stage is trained using only masked L1 losses. The second stage uses GAN loss to refine the coarse image. It requires proportionately complex and lengthy training procedures due to its two-stage nature. After [1] work, many studies aim to

solve the problems encountered in *SAAHPT* and produce more realistic images.

If the pose to which a person is intended to be transformed is very different from that person's pose, it is difficult for the neural network to make this transformation directly. This is because the neural network consisting of convolution layers learns the local relationship. This situation causes some views to be misaligned due to pose differences. To alleviate this problem, Siarohin et al. [13] introduce deformable skip connections that modify the foreground human pose in the generator. This method requires extensive affine transformation computation and is sensitive to inaccurate pose disjoint. This situation is resulting in poor performance for some rare poses. After this work, many articles address the problems caused by the difficulties of human pose and pixel-to-pixel misalignments. Similar to [13], [3, 34] integrate affine transformations into their models to alleviate some of the problems encountered during human pose transfer. These problems are that human has too much deformation due to the articulation of the human being, and the person images of different poses under different views contain great differences in appearance.

While most of the approaches that exist in the literature apply conditioning in one direction (creating the image at the desired pose using only the input person image), AlBahar et al. [87] apply the conditioning in both directions, ensuring the flow of information from the input person image to the guide person image in addition to the existing flow to increase image generation success. In addition, AlBahar et al. propose a novel normalization technique called the feature transformation (FT) layer.

Moreover, unlike the one-stage transfer methods are applied in other studies, Zhu et al. [14] propose a progressive pose attention transfer network called PATN that transfers the pose locally at each stage in order to provide realistic generation in situations that require large pose transfer such as transforming a sitting person into standing.

2.3.2 Random Human Pose Transfer (RHPT)

This approach disentangles information such as the scene, the person's appearance, or pose, providing control over one or more of them. Ma et al. [4] separate the image into foreground, background with human masks and designs a different encoding

structure for scene, appearance, and pose. In this way, this approach makes each of them controllable over separate representations. Then, this method combines the features extracted by the encoders into an image through the decoder network. In this way, Ma et al. provide new images to be produced in arbitrary scenes, appearance or poses by changing any of these representations. While [1] allows the synthesis of images of the person in any arbitrary pose, [4] also reproduces the human body and the background image from a noise distribution represented in latent space. Likewise [4], Essner et al. [16] aim to disentangle appearance and pose. To this goal, they present a U-Net that maps from shape to target image, conditioned on a VAE latent representation for preserving appearances. It leads to misalignment of several views as appearance features are challenging to be represented by a latent code with a fixed length. In [4, 16], although the controllability of different features such as pose, appearance, background is increased, the quality of the generated images decreases. In addition, while these types of approach synthesize the person images in any random background, in our study the background that is desired to be generated can be given as a condition.

2.3.3 Source Background Aware SAAHPT (SBA-SAAHPT)

Studies in this approach aim to preserve the appearance of both the person and the background while transferring the pose. Balakrishnan et al. [2] , whose study is one of the works in this field, produce realistic images by filling the changes in the background that are observed due to the different poses. It takes the source person's image and pose as well as the target person's pose information as input. This work separates the source person into foreground and background using the pose information and uses this information as a mask to spatially move the body parts to target locations separately. While achieving this purpose by using different U-Net based reconstructions deep network for scene and view, GAN prevents blurry image formation thanks to its deep network.

2.3.4 Target Background Aware SAAHPT (TBA-SAAHPT)

Many human pose transform methods focus on generating person images conditioned on a desired pose, while the generated backgrounds are often blurred. There are some approaches that produce background. It is one of these studies at PCGAN. This study attacks some of the problems observed in the PG2 method. The rendering process in PG2 is designed as a black box consisting of an encoder and decoder, and the pose and source image is combined on channels and delivered to the network. For this reason, it is difficult to further organize the lower body parts. To alleviate this problem, [34] uses a network consisting of two encoders (E1, E2) and one decoder. One of the encoders aims to extract the latent representations of the source appearance feature, the other extracts the pose feature and the background feature. E1 is fed with the source image and its landmarks. Meanwhile, E2 is fed with the target image and also its landmarks as well as background. Additionally, this work also splits out the foreground person using the extracted mask of the source images to better match the appearance of the foreground person to the background. The masks of both the source and the target image are extracted using the Mask-RCNN trained on COCO2014. And discriminator fuses the feature maps extracted from E1 and E2 by U-Net structure. This way the foreground and background are brought together.

In addition to the pose points, masks are also used in both testing and training stages. Since there are no hand-labeled masks in the datasets used for Re-ID, mask-extraction networks spefically Mask-RCNN are used. For the reason that the mask obtain network is trained on the Coco dataset, and there is no similar viewpoint as the Market dataset, its performance is not high. Especially if one or more joints are occluded by another object or by itself, the network can produce foreground masks at these pose points and mislead the person generation success in occluded people. In addition to this drawback, when we examined the results produced, we observed that the generated people missed the local foreground details.

Another example of these studies is [3]. Similar to PCGAN, this study learns the foreground and background with separate encoders, and unlike PCGAN, it uses only the joint information to obtain the masks and places a rectangle at a certain angle around the relevant joint point, depending on the person's width. As in [34, 13]

studies, it increases the success of pose transfer in cases where the pose points of the target and source image are far apart, thanks to its affine transformation. This study uses the PRW data set and takes the target image to place a person in a target scene. For this purpose, instead of taking the scene with the target person, it uses a non-human scene and learns to place the source person in that scene. In this study, if the lighting conditions of the scene is different from the source person, it was observed that the brightness of the person's appearance was adjusted according to the background when placing the person. Unlike this study, we emplace the source person in a scene where the target person is located and while doing this, we fill the gaps caused by the different body structures of the source person and the target person in a realistic way.

CHAPTER 3

METHOD

In this section, we present our method for generating person images with desired pose and scene. In the Section 3.1, after briefly giving information about our purpose and the problem we are trying to solve, we talk about our approach to solve this problem in the Section 3.2. We explain the dataset we use and how we created this data set in the Section 3.3. After presenting our architecture design in detail in Section 3.4, we conclude with explaining our training stages and loss functions in Section 3.5.

3.1 Problem Definition

Although there are many studies in the literature, such as PG2 and PG2-inspired studies that produce a person in the desired pose, there are few studies that put a person in a particular scene in the desired pose. For this reason, we present an approach that attaches importance to the background of the scene as well as protecting the appearance of that person while producing a person in a different pose. For this, instead of producing a background that can be controlled depending on its distribution, we aim to produce according to the background of the target person given by the user as with the pose representation. As an example of generating a person's image according to another person's background and pose, we can demonstrate this as an example of observing/watching an actor in a scene that was initially recorded with stunt. For this purpose, our network takes the picture of the stuntman who has the information of the scene and pose to be produced as a target, as well as the actor to be produced in a different pose. Using this information, we aim to emplace the actor on the stage where the stuntman is, as well as producing the actor in the pose of the stuntman.

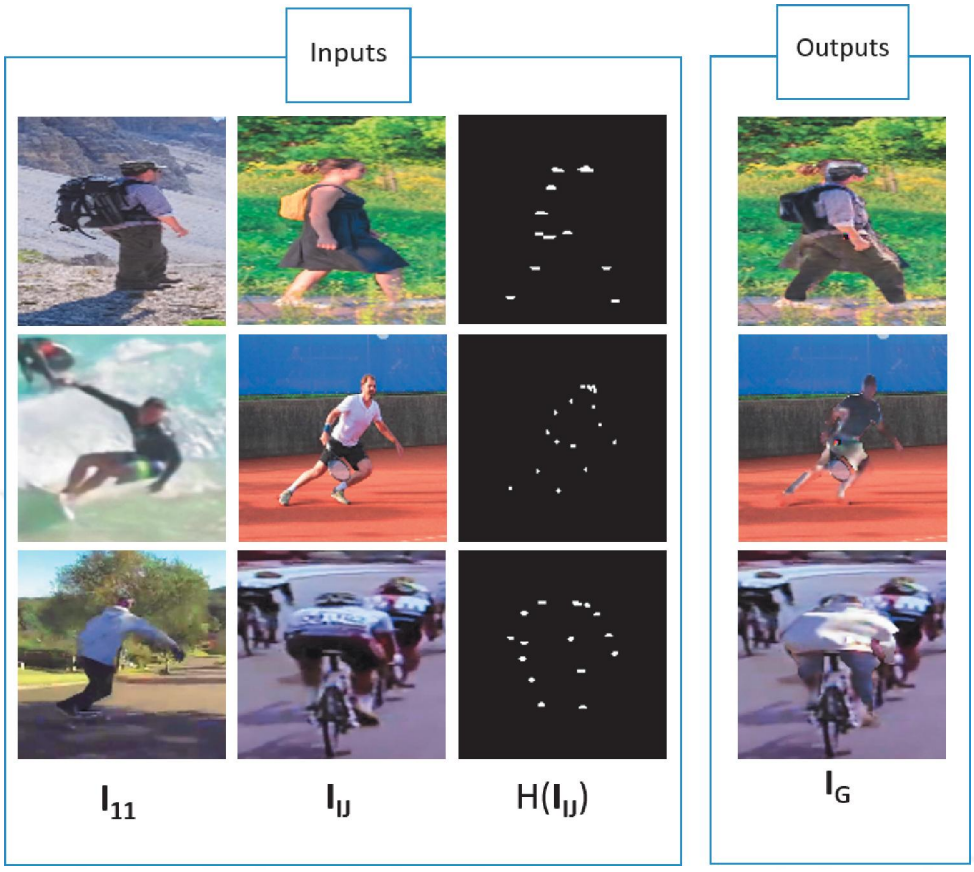


Figure 3.1: Showing inputs and outputs of our approach. The first column represents the actor, the second column represents the stuntman, and the third column represents the stuntman's pose heatmap. Our approach uses the pictures in the first three columns to produce the image shown in column 4, which has the appearance of the actor and the pose and background of the stuntman.

The inputs and outputs of our approach are given in Figure 3.1. In this figure, the source image, actor, is represented with I_{11}, the target image, stuntman, is represented with I_{IJ} and the pose information we have obtained using pre-trained pose estimation network over the image of the target person image is also represented by $H(I_{IJ})$. In this context, our goal is to emplace the actor is represented in I_{11} on the stage shown in I_{IJ}. In doing so, we want to sharply reconstruct both the background and the pose as well as the foreground information that was not available in the PG2 study.

3.2 Proposed Approach Overview

In our approach, we are inspired by the architecture of the PG2 [1] network to reproduce a person in another pose and make some changes to that network to place one person in another person's scene. Our model consists of a combination of GAN and U-Net architectures. Also, we take advantage of the strength of the join of the pixel reconstruction loss function used in the Autoencoder architecture and adversarial loss function used in the GAN architecture.

GAN architecture consists of two components, the generator and the discriminator network. In this thesis, while the generator network produces a person in the desired pose and background, the discriminator network makes the image produced by the generator more realistic by comparing the generated image with the desired image in the dataset.

We use a generator (G) network designed to emplace the actor's image on the stunt's scene while the actor into the stunt's pose, at the same time without distorting the background of the stuntman. This network is based on the U-Net architecture that we mentioned in Section 2.2, which ensures the preservation of the fine-grained image through skip connections, and it produces a desired person's image as output by taking a group of images as input. These inputs are pictures of the actor and the stuntman and the pose information of the stuntman. As in other pose transfer approaches, the image of the source person and the pose information of the target person is fed to the generator network, while we also use the image of the target person, namely the stuntman, in order to produce the desired background.

We use the discriminator network, another GAN component, to produce more realistic images by ensuring that the image created by the generator network converges to the desired image in the dataset. In addition to enabling the generator network to produce images similar to the images in the data set, Discriminator distinguishes the real and produced data using the data contained in the data set and the data created by the Generator, as mentioned in 2.1.3. For this purpose, the discriminator network evaluates the image/s produced by the generator as fake, and the images in the dataset that want to be produced as the desired as real images. For this reason, we feed the

image of a person produced by our generator network in a different pose and background as a fake image into our discriminator (D) network. In addition, we randomly select one of the images of the actor or stuntman and feed it into our discriminator network as a real image. There are two reasons why we randomly select a person's image and feed it into the network. The first reason is that we do not have the expected real image as in other studies. In other studies, the network is trained over pictures of a person in different poses, and these pictures are included in the dataset in pairs. In our approach, it is necessary to have images of two different people in similar and different poses and there is no such dataset in the literature. For this reason, we do not have the desired image to feed into the discriminator network. To solve this problem, by providing the discriminator network with images of real persons, we enable the generator network to learn persons and scene information, and also to make decisions in the discriminator network based on this information. The other reason is that when we only give the image of the target or source person, our network tends to produce similar images because it cannot analyze a wide variety of images and it tends in a single direction because it will either only see the source person and the background, or only the target person and background.

In the vanilla GAN architecture, the discriminator network is the decision-making mechanism and is responsible for updating both its own and the generator network. The Discriminator network updates its parameters, aiming to prevent the generator network from fooling the discriminator network in future iterations. It also updates the parameters of our generator network to ensure that the images produced by our generator network are closer to real people and scenes. In this way, only in adversarial trained networks, the generator network learns how to produce pictures according to the information from the discriminator. The generative network learns how to produce images according to the information from the discriminator network by taking a series of inputs, and the discriminator network states that it should teach a randomly selected human so far. Therefore, we need to lead our U-Net network, which we use to reconstruct one person to reproduce another person's pose and scene, with pixel-wise reconstruction loss functions as well as adversarial loss. In this way, we direct our generator network by comparing a person with the targeted pose and background image, and by combining reconstruction and adversarial training, we benefit from the

strengths of both.

Other pose transfer methods have a combination of the image of the source person and the image of the source person in the target pose during the training phase. This information enables the network to accomplish the task by defining the reconstruction loss function between the image they produce and the expected image. Therefore, as in other approaches, we cannot define the reconstruction loss function by directly measuring the similarities between the produced image and the desired image, as we do not have image pair information in different poses belonging to the same person with the desired scene.

To solve this situation, we need an image with the actor in the same pose as the stuntman. From Davis and Youtube video datasets, details of which are given in Section 3.3, we can obtain images of the actor in different poses. Since these datasets are taken with moving cameras, they can change constantly as a person's background. However, there are no images of the same person taken with different videos. So another image of the actor in the same pose as the stunt is in the appearance of the actor given to the network as the source and his/her background image may or may not be the same as the actor in the different pose, but certainly not the same background as the stuntman. Since the dataset does not include an image of the actor with the same pose and background as the stuntman the pixel-based reconstruction loss function cannot be directly defined on these images.

To alleviate this problem, we use foreground and background masks. The details of the mask loss function are explained in Section 3.5.1. We used the power of people's masks to put the source person in the target person's pose and without disrupting the target person's scene. Therefore, we extract and combine appearance features of the actor, pose features, and background features of the stunt to synthesize the desired images in the generator network in training time. We use the mask information in loss definition and this information is not needed during the testing phase. In this way, different datasets can be used in the test phase.

The overall framework is shown in Figure 3.2. In this figure, the actor is represented with I_{11}, the stunt is represented with I_{IJ}, and I_{12} shows the actor in the same pose as the stunt. Although the I_{12} and I_{IJ} pose information is the same, the foreground

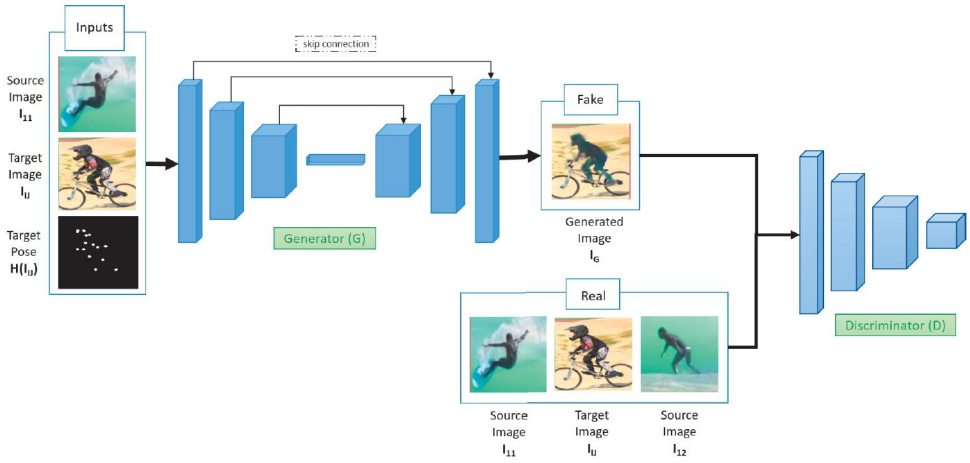


Figure 3.2: The overall framework of the proposed scene-preserving person appearance transfer network with its inputs and outputs.

and background are different. Our goal is to sharply reconstruct both the background and foreground information while emplacing the actor represented in I_{11} on the stage shown at I_{IJ} with his pose. Here, the generated image is shown as I_G. In the I_G image, the foreground information comes from I_{12}, the pose information comes from I_{12}, and the background information comes from I_{IJ}. In addition, the pose information can come from the I_{IJ} because the poses of I_{12} and $_{IJ}$ are the same.

The model is trained on paired image datasets $\mathcal{X} = \{\left(I_{11}^{\{i\}}, I_{IJ}^{\{i\}}, H_{IJ}^{\{i\}}\right)\}_{i=1,..,N}$, which contains the images of the actor and the stuntman and the pose heat map is represented with H_{IJ} derived from the image of the stunt described in Section 3.4.1. In addition, we focus on the actor's appearance and the stunt's scene individually, using foreground and background mask information as the loss function in the training phase, details of which are described in Section 3.4.1. In the test phase, we use the triple information fed to the generator network similar to the training phase, but we do not use the masks used in the loss function.

In addition, since the Market and Fashion datasets, the details of which are given in Section 3.3, could not provide enough information for our problem, we conducted our trainings on Youtube VOS and Davis datasets.

3.3 Dataset

Market and Deep Fashion datasets. Pose transfer networks use pairs of images of persons in different poses, and Market and Fashion datasets are preferred to reproduce a person in a different person's pose. Both datasets contain images of multiple persons in different poses. In the Market data set, there are 32,668 images of 1,501 different people collected over 6 different cameras, and 439,420 pairs were obtained using different images of the same person. The pairs amount in the Fashion dataset is 146,680.

Although these data sets are rich in images, they cannot be used in our approach for two reasons. The first of these reasons is that both datasets do not have diversity in terms of background. The Fashion dataset does not contain any information about the scene, as it is pictures taken in the studio. Although the Market dataset is taken from 6 different cameras, there is not much variety in the background due to the cameras looking at a similar scene. The second reason is that since both datasets are Re-ID datasets, they do not have hand-tagged masks. Two different methods have been used to obtain the masks. The first method uses the pose information and the human joint topology knowledge to fill the between joints, while the second method uses the MaskRCNN trained on the COCO dataset.

In both methods, it is observed that the mask cannot be obtained well, especially in some people with one or more pose joints that are occluded. This situation negatively affects the success of our network, which learns to manipulate and fill pixels by using the loss function over the foreground of the source person and the background of the target person.

Youtube VOS and Davis datasets. Since the Market and Fashion datasets are not suitable for our task, Youtube VOS [93] and Davis [94] video datasets, which are among the segmentation datasets with masks, are used in our studies. Besides both Davis and Youtube datasets have higher-quality video resolutions, they also have complicated video content with multiobject interactions, occlusions, and camera motion. Davis has 150 videos with multiple objects per video. This dataset consists of totaling 10459 annotated frames and 376 unique objects. Youtube VOS has 4,453

Figure 3.3: Visual examples of images of people in different poses with high-frequency foreground and/or background taken from Youtube VOS and Davis datasets. These datasets include wide-ranging poses, human appearances, and scenes that reflect the real world.

YouTube videos with 94 semantic categories covering humans in various activities (e.g. tennis, skateboarding, motorcycling, surfing), animals (e.g. butterfly, cat, horse), vehicles (e.g. train, bicycle, truck, sedan), and common objects (e.g. table, plant, knife, ball, hat). This dataset consists of totaling 197,272 annotations and 7,755 unique objects. In this dataset, videos with images of 1702 different people engaged in different activities to increase the diversity of human movement are tagged.

In these datasets, there do not exist any well-defined rules or limitations that can

not represent the real-world scenarios well and can not favor of neural networks. In contrast, these datasets contain a large set of poses and scene that reflects and represents the real world better than sample datasets mostly used in literature. In Figure 3.3, we present examples of different people in different poses taken from Youtube VOS and Davis datasets. In addition, unlike the datasets frequently used in human pose transfer, these segmentation datasets are taken from very different perspectives and have a wide variety in terms of human body structure, illumination conditions, and image distributions.

Besides these advantages, it has two disadvantages. The first of these disadvantages is that although these datasets have different poses of a human being on video, some parts of the video have a lot of human mobility and some parts are not. This situation causes the same person to repeat in a similar pose, and such pictures need to be cleaned while creating pairs. Another disadvantage is that due to these datasets have complex scenes witnessed in the real world, partial or full occlusions are observed. The amount of occlusion is important because the desired results cannot be obtained in cases where too many joints are occluded in human image generation networks.

In order to solve these problems, we applied some preliminary processes to the datasets. We also adapted these datasets to obtain the necessary input pairs for our architecture during the human pose and background transfer.

Dataset elimination and creating pair.

Re-ID datasets are generally used in human pose transfer, and generally, these datasets include the following criteria (i) people are in the center of the picture and the majority of the pixels of the image contain information about the human appearance, (ii) the picture has a specific resolution, (iii) the amount of occluded pose not higher than a certain value. In addition, in order to create a person in another pose, the pose information and human image pairs in different poses are needed. In this thesis, after pre-processing the Youtube and Davis datasets, which are the segmentation datasets, we make some calculations in order to obtain the input picture pairs required for our design.

First of all, our datasets include not only people but also objects belonging to different

classes because these datasets are based on video segmentation. For this reason, only videos containing human images should be selected from the datasets. In addition, since the dataset is not designed solely on human pictures, the person image usually occupies a small area of the scene. In order to learn the foreground and background in a balanced way, we crop the image in a window that we have determined depending on the width of the person. This cropped image contains background information as well as the person's picture. Also, because some video sets have more than one person, we evaluate each person in the video individually. Then, we extract the coordinate information of the joint points for each person in all the videos. We use the pre-trained human pose estimation (HPE) deep neural network, which we will detail in the following sections, to obtain the joint points. It has some localization errors, false alarms, and missing predictions because this network is trained in another dataset. We use the person's mask to correct some localization errors. Since our dataset is a segmentation dataset, the reliability of the masks is high. We determine if there is a localization error by examining whether each pose point obtained from a network overlaps with the mask. If the joint point with localization error is in a certain neighborhood of the mask depending on the size of the person, we shift the joint point so that it is inside the mask. If the joint point is out of the neighborhood of the mask, we categorize it as a false alarm. Figure 3.4 shows the corrected results for some of the localization errors caused by HPE in individuals with different poses. If the sum of false alarms and missing predictions is more than 3, we do not include that picture in our training or test set. In this way, we ensure that the joint points that are occluded more than a specific number are removed from the dataset, as well as the partial body image. We show examples of false alarms observed HPE in Figure 3.5.

Deep neural networks work with specific input sizes, and pictures with different image sizes than this input size reduce the success of the network. We also perform different eliminations in order to increase the success of the network. The first of these is that our network is designed to work on square images, so we eliminate images with an image aspect ratio of 2.5 from the dataset. In Figure 3.6, we give images with aspect size greater than 2.5. When these images are given to the neural network trained on square images by rescaling, the properties of the objects become lost and

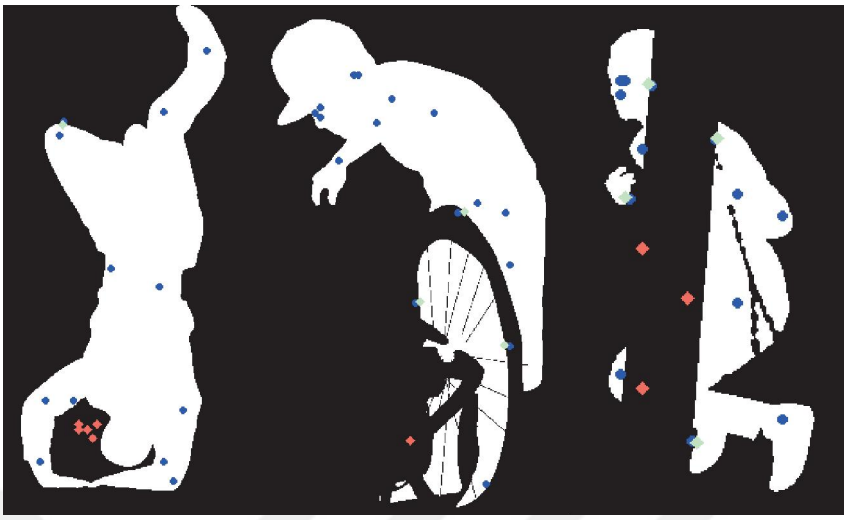


Figure 3.4: Visual examples of corrected results of some of the localization errors caused by Human Pose Estimation (HPE) in individuals in different poses. The blue dots show the HPE results, the green dots indicate the new joint location obtained by correcting the pose points close to the mask from the points outside the mask caused by the localization error of HPE, and the red dots indicate the uncorrected results due to the localization error being far from the mask.

.

the network cannot produce meaningful information from this image. Another elimination process is that we perform when the size of the images we receive as input is smaller than half of the input size of our network. In addition, if the person in consecutive video frames is not moving, we remove the frames without movement from the dataset in order to prevent the distribution of the dataset due to the same person repeating in a similar pose. Since people do not move much in some scenes of the video, we calculate the Mean Square Error (MSE) of the poses of the people in successive scenes to prevent the same image from being processed and if MSE is below a certain value, we don't include that training dataset.

After performing elimination operations on our data sets, we combine the two data sets to increase the variety and amount of data. After that, we obtain the necessary inputs during the pose transfer training. For our approach, when creating the pairs, the image in which the actor is in the same pose as the stuntman, in addition to the image in which the actor has two different poses is also required. Images of the actor

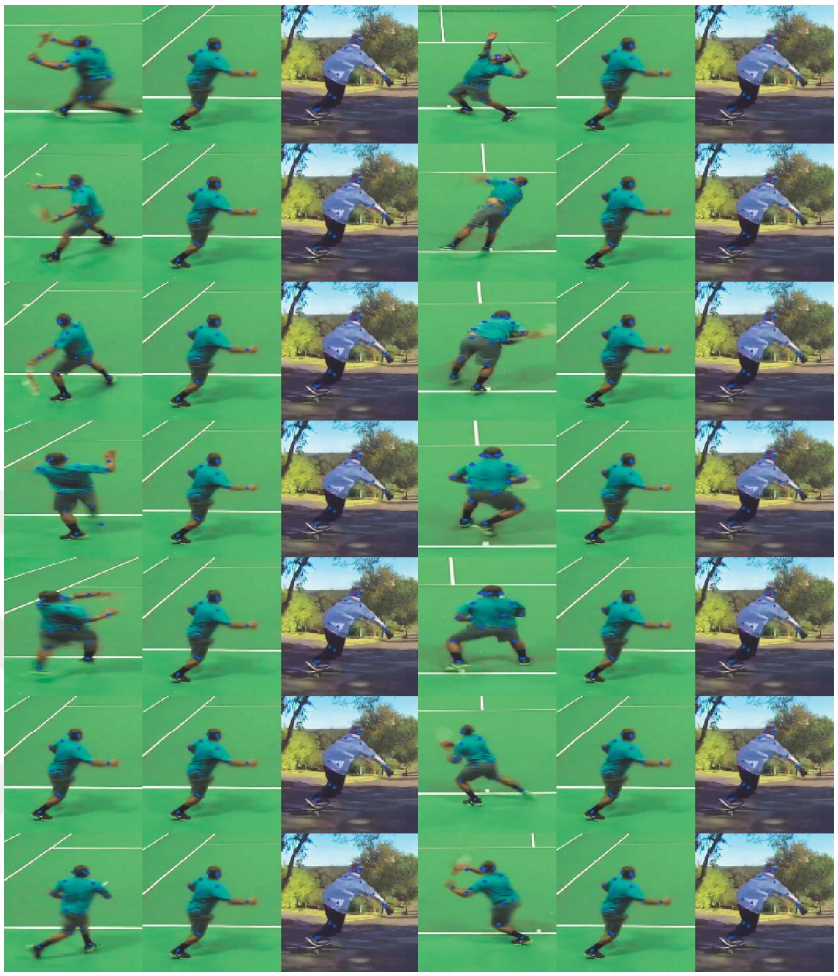
Figure 3.5: Visual examples of people eliminated from the dataset due to the occlusion of more than 3 or 3 joint points or because some pose points are not visible due to partial body structure. Due to the partial body structure of the picture of the girl in the first column, there are not 5 joint points. Due to the excess localization error in the face area of the boy in the second column caused by the pre-trained HPE network, 6 joint points do not overlap with the mask. Since the legs of the lady in the third column are covered by another object, 4 joint points do not overlap with the mask.

.

Figure 3.6: Visual examples of people eliminated from the dataset due to the aspect ratio of the images greater than 2.5.

in different poses can be obtained through the video dataset. However, it is necessary to find images where the actor is in the same pose as the stuntman. Therefore, for each person, we store the pictures of a given person as a duple with his/her coordinate information of the joint points. These stored picture and joint information duple for each person are used to calculate the euclidean distance to all other persons' picture and joint information duple in order to detect the humans who have similar poses so that the pose of the actor and the pose of the stunt can be used to mask their foregrounds that will feed the loss function.

While we use the images of the actor and the stunt in a similar pose for pixel-based comparison during the training, we use the images of the actor and the stunt in dif-



(a) (b)

Figure 3.7: Showing an example of dataset pairs. The first two columns are images of the actor in different poses and are derived from different video frames in the video set. The third column is the picture of the stuntman with the same pose as the actor in the second column and is obtained by calculating the MSE from the pose information of people in different videos to find people in similar poses. Columns in (b) are pictures of the columns in (a) in different poses of the same actor.

ferent poses to feed into our generator network. For this purpose, we need 3 different person images during the training phase and these are the image of the actor in a different pose with the stuntman, the image of the actor in the same pose with the stuntman, and the image of the stuntman as shown in Figure 3.7. We use our mech-

anism of detecting people in similar poses that we mentioned above to get images of the actor and the stuntman in similar poses. Since our dataset is a video set, we use other images of the video to find images in which the actor has a different pose than the stunt performer.

We create our pairs over the combination of Youtube VOS and Davis datasets to increase our pair amount and diversity our dataset. We separate the triples we have created as a result of these processes for our training and test set. We divide our test set into 3 different categories. The first of these is a pair of 20 different people's images in different poses, and these people have no images in our training set. We call this category test-test set, which consists of 76 different triads belonging to 20 different people. We use this category to measure the success of our network in people who have never seen it. The second category is, there is no image of one of the actors or stuntmen on the training set, while the other has images in different poses on the training set, but the image in that pose is not included in the training set. Our test-train data set, the second category consists of 1088 different triads belonging to 26 different individuals. In the third category, although the actor and stunt do have images in different poses in the training set, the images in a particular pose are not included in the training set. In our third category, the train-train category is used to measure the success of our network in images where it is trained over human appearances and scenes similar to the distribution of our network. This category consists of 3048 triplets belonging to 40 different persons. In addition, we created the train-train set to include such triads to investigate how our network will behave in people with different body structures.

3.4 Network Detail

In this section, after we have given some preliminary preparations which are pose heatmap, mask and normalization technique, used by our method with their notations, we describe the architectures of our generator and discriminator.

3.4.1 Preliminaries

Pose heatmap. In Davis and Youtube datasets, there are no tagged joints of people that contain pose information. To avoid the tedious pose annotation, we use a recently published pose estimator [95] to obtain the joints of the human body such as hands, elbows, feet, etc. The pose estimator produces the coordinates of 17 joint points. $P(I_{IJ}) = (p_1, ..., p_k)$ is a series of k 2D points that describe the positions of human-body joints in the target image. k is the maximum joint point obtained from the method [95]. There are location errors, false alarms and body parts due to the fact that the training set and the test set are different datasets in the joint points obtained by human pose estimation (HPE). Some of the false alarms that occurred in HPE were corrected according to the mask information.

We use these pose points in both training and testing phases to create the pose heatmap. Heatmap holds each joint in separate channels and has as many channels as the joint point and meantime is a 2D matrix of the same dimension as the image. Due to this separation structure, it is preferred more than the stickman. Each channel of the heat map is calculated by the Equation (3.1).

$$H_j(p) = \exp\left(-\frac{||p - p_j||}{\sigma_2}\right), (1 \leq j \leq k) \tag{3.1}$$

p_j denotes the $j - th$ joint location. The σ value is chosen as 6 pixels according to [34] paper.

Mask. In addition to training our network as an adversarial manner, we want to take advantage of the pixel-wise reconstruction loss function. As described in Part 3.2, we have actors and stunts in similar poses, represented by I_{12} and I_{IJ}, respectively. Although the foregrounds of these two people are similar due to the close positional information, they do not completely overlap due to the different body structures of the people as shown in Figure 3.8. In order to prevent errors that may arise from this, we calculate the intersection of the foreground masks of the two people given in the Equation (3.2) as the final foreground mask. Since our dataset is a segmentation dataset, people have foreground masks and the value of the pixels where the human

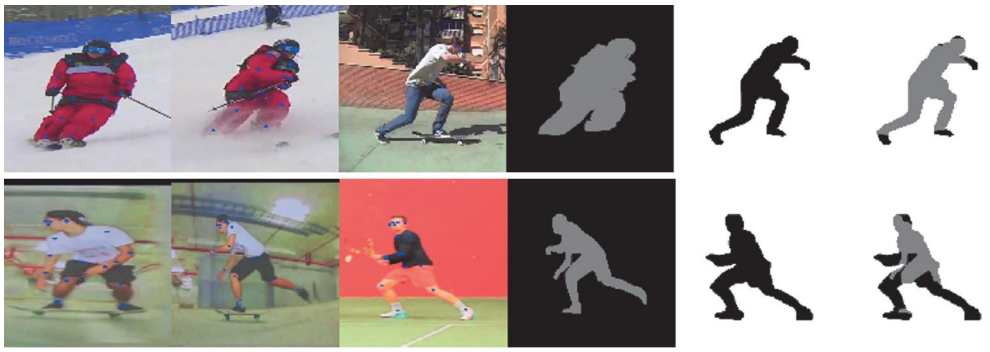


Figure 3.8: Visual examples showing that two different people in the same pose do not have the same masks due to their different body structures. 1st column is the picture of the actor, 3rd column is the picture of the stunt, 2nd column is the picture of the actor in the same pose as the stuntman. The 4 columns are the mask of the 2nd column (the actor in the same pose as the stuntman), the black area represents the background, while the gray area represents the foreground. The 5th column is the mask of the 3rd column (stuntman's foreground mask). Column 6 is the intersection of the actor's mask shown in gray in column 4 and the stunt's mask shown in black in column 5.

is 1, while the places that are not are 0. We created a similar mask using the Equation (3.3) for the background scene.

$$\mathbf{FM} = M_{I_{12}} \wedge M_{I_{IJ}} \tag{3.2}$$

$$\mathbf{BM} = (1 - M_{I_{12}}) \wedge (1 - M_{I_{IJ}}) \tag{3.3}$$

When the mask of an actor is weak or short compared to the stuntman, we observed that there are gaps between the foreground and background masks that neither of them has any knowledge of. It significantly fills such gaps for adversarial loss.

Normalization. Batch normalization, one of the normalization techniques prevents modal collapse of deep generative models by producing less variety of results by the manufacturer, normalizes the input properties of a layer to have zero mean and unit variance. Apart from this, it also offers a solution to the problem of insufficient

parameters at the beginning of the training phase. Apart from these advantages, there is a problem of not being used in the test phase. We observed this situation during our own tests, and as a remedy, we either turned off these layers during the test phase or chose the batch size parameter as 1.

Unlike batch normalization, the **instance normalization** layer that calculates the mean and variance for each channel of each sample can be applied at test time.

Spectral Normalization, which is preferred as a solution to the mode collapse and exploding gradient problem often observed in discriminator networks, controls the Lipschitz constant. It has less computational burden compared to WGAN_GP. Having a global editing technique, this method is easy to incorporate into existing applications.

3.4.2 Architecture

Generator. We use U-Net architecture to integrate the actor's picture, the stunt's picture and the stunt's pose heatmap, and then generate the picture of the actor on the stunt's stage. By combining I_{11}, I_{IIJ}, and I_{HIJ} on a channel basis, we feed it into our U-Net architecture. This architecture consists of encoder and decoder subnets. The first convolutional layers of the encoder network integrate I_{I11}, I_{IIJ} and I_{HIJ} from small local neighborhoods, while the deepening layers integrate the body parts. In this way, appearance and scene information can be integrated and transferred to neighboring body parts. At the end of the encoder network, we use the fully connected layer so that remote body parts can share information among themselves. In the case of using a convolutional layer instead of a fully connected layer after the encoder last layer, the network cannot achieve the expected success when images of people with very different poses are wanted to be produced.

After that, we used decoders to generate an image using the integrated information. this network consists of a series of deconvolution layers symmetrical to an encoder.

We prefer this model to be deep and powerful, as our generative model needs to learn the distribution of the dataset and to produce pixels according to different people

and scenes. As with the PG2 approach, we take advantage of the power of Residual neural Networks. We use the ResNet in every layer except the first and last layers in our generative network.

In addition, after each convolution layer, we apply the normalization and activation functions respectively. Although we tried different approaches as normalization, we observed similar results Instance Normalization and Batch Normalization after fixing with certain batch size. We prefer instance normalization for our later trainings due to the fact that batch normalization had to be turned off during the test phase and the results obtained depending on the number of batches as stated in Section 3.4.1.

PG2, which we are inspired by, consists of two different U-net-based generator networks. After training the first network with the loss function only, similar to the auto-encoder network, a second generator is trained with both reconstruction and adversarial loss function using the outputs of the first network. PG2 also does not apply normalization after convolution layers and all activation functions are Relu. We use the first generator network of the PG2 study with both reconstruction loss and GAN loss function. We are also changing the activation function as well as adding the normalization layer.

Discriminator. The purpose of the discriminator network is to decide whether the images produced by the generator have a real scene and human appearance. Since the desired image is not directly available in datasets, this image is obtained with masks. This network has not been trained conditionally because of the gaps in people's body structures. The discriminator randomly selects the real image of the actor, the picture of the actor in the stunt pose, or one of the stunt's picture.

In this work, we follow the DCGAN structure as the baseline generative adversarial network. DCGAN uses inverse convolutional layers (also called as deconvolutions) with stride convolutions to upsample the tensors to the image size in the generator network. They map the latent input vector of the generator to a 5-dimensional tensor by applying an affine transformation. In the discriminator network, they use convolutional layers with strides to eliminate the pooling operations and allowing the network to learn its own spatial downsampling. The last convolutional layer is flattened and fed to affine layers for the classification output with sigmoid function. Batch nor-

malization is also applied for convergence issues and poor initializations for better flow of the gradients to the deeper layers except two layers. These are the generator's last layer and the discriminator's first layer. Relu activation function is used in the generator network whereas leakyRelu is applied in the discriminator network.

Unlike the normal DCGAN structure, we use the spectral normalization specified in Section 3.4.1 to avoid the mode collapse problem. Moreover, DCGAN uses cross-entropy for binary classification as a loss definition, so it uses sigmoid in its last layer. Since we use the hinge loss function, we are deducting the sigmoid function.

Also, in order to prevent the mode collapse by remaining stuck to the local minimum observed in the separator, which we explained in Section 2.1.3 , the approaches that the discriminator is updated more than once or the discriminator's loss function multiplier is greater than the generator is used. Among these approaches, we preferred the TTUR approach in order not to slow down the training phase.

3.5 Training

After the generator and discriminator network are trained together, only the generator network is used in the test phase. Therefore, some inputs used by the discriminator network and loss functions parameter are not required during the test phase. Since only the generator model $G(I_{11}, I_{IJ}, H_{IJ})$ is used in the test phase, the image of the actor (I_{11}), the image of the stuntman (I_{IJ}) and the pose heatmap (H_{IJ}) obtained after extracting the pose information from the image of the stuntman are needed to obtain the desired image.

We use 2 different loss functions to produce a person in the desired scene and pose. Using the Adversarial loss function during the training, we ensure that the images produced by the network have a real scene and human appearance. In addition, using the Pixel-wise Reconstruction Mask Loss function, we ensure that the result produced looks like an image that has both the background and pose of the target person and the appearance of the source person. Each mainstream loss function is summarized in the following subsections with the formula.

3.5.1 Pixel-wise reconstruction mask loss

We use foreground and background masks, which are described in Section 3.4.1, to produce the desired image using image information of different people (the appearance of the actor in a different pose and the background of the stuntman). The image of the actor in the same pose as the stunt, obtained from the video set, is one of the basic building blocks for our loss function. The actor's appearance in the target pose is the view of the result we want to achieve. For this reason, in this thesis, we calculate the pixel-wise mean square error (MSE) as reconstruction loss function using a foreground mask over this picture of the actor:

$$\mathcal{L}_{FG} = ||FM \odot I_{12} - FM \odot G(I_{11}, I_{IJ}, H_{IJ})||^2 \tag{3.4}$$

Similarly, the scene of the stunt given as the target is the scene we want to produce, and we calculate another reconstruction loss function by applying a background mask to the stunt's picture:

$$\mathcal{L}_{BG} = ||BM \odot I_{IJ} - BM \odot G(I_{11}, I_{IJ}, H_{IJ})||^2 \tag{3.5}$$

Our cummulative reconstruction loss function is a weighted sum of above terms:

$$\mathcal{L}_{l2}^{Mask} = \lambda_1 * \mathcal{L}_{FG} + \mathcal{L}_{BG} \tag{3.6}$$

Also, we implement a different loss function to check the effectiveness of our intersection mask, which we define to minimize the effects on human structural differences. Before the training phase, we interpolate the gaps caused by these structural differences between the images of two people who were pair, filling with the background. For this, we used Opencv's warping function. If the body types of two people are very different from each other or if one is closer to the camera, the interpolation process cannot achieve the desired success due to the excess amount of space formed. In such cases, we leave the task of the image to have a realistic appearance and to close these gaps to the discriminator network. As a result, our reconstruction mask loss function changes into

$$\mathcal{L}_{l2}^{withoutMask} = ||I_{waping} - G(I_{11}, I_{IJ}, H_{IJ})||^2 \tag{3.7}$$

where I_{waping} represents the image obtained by Opencv's Warping function. The reason for this is that we have a rough image of the person we want to produce as a target.

3.5.2 Adversarial loss

A discriminator is a binary classifier that decides whether the incoming image is a real or a produced image. The cross-entropy loss function is used in the vanilla GAN approach and different loss functions are proposed because this loss causes the vanishing gradient problem. Hinge's loss function is one of them, which uses a binary classification structure with values between $\{-1, 1\}$. Hinge Loss makes sure that the actual images are between 0 and 1 and the generated images are between $\{-1, 0\}$. It increases the classification ability of the discriminator by increasing the amount of punishment when there is a difference in the sign between the actual or produced class values. Discriminator Hinge loss is given by:

$$\mathcal{L}^{D}_{hinge} = \mathbb{E}_{I \in \mathcal{X}}[\max(1 + D(G(I_{11}, I_{IJ}, H_{IJ})), 0) + \max((1 - D(I)), 0)] \tag{3.8}$$

where real image is chose at random from $\mathcal{X} = \left(I_{11}, I_{IJ}, I_{12}\right)$ for discriminator update.

Generative adversarial loss tries to maximize the following function:

$$\mathcal{L}^{G}_{hinge} = \mathbb{E}_{I \in \mathcal{X}}(D(G(I_{11}, I_{IJ}, H_{IJ}))) \tag{3.9}$$

We train your adversarial with the TTUR mechanism. This is why the discrimanator and generator initial learning rate are different.

3.5.3 Total loss

We trained our network with adversarial and reconstructions loss.

The total reconstructions loss equation:

$$\mathcal{L}^{total\_recons} = \lambda_2 * \mathcal{L}^{Mask}_{l2} \tag{3.10}$$

And total adversarial loss equation:

$$\mathcal{L}_{hinge}^{total\_adv} = \mathcal{L}_{hinge}^{G} + \mathcal{L}_{hinge}^{D} \tag{3.11}$$

In the next chapter, we present a detailed experimental evaluation of our approach.

CHAPTER 4

EXPERIMENTS

In this section, we present and discuss our experimental results. First, we explain our training process by detailing the structure of our network. We then present the results of the different categories of experiments we conducted visually and share our comprehensive reviews on successful and unsuccessful scenarios. Finally, we present the evaluation methods used in pose transfer and the results we obtained in terms of them.

4.1 Implementation Details

U-Net based generator network with skip connection property consists of two subnets that are encoder and decoder. From the knowledge of the auto encoder structure, the decoder network is exactly reverse of the encoder network. The encoder network consists of one initial block group, K ResNet blocks that are depending on the size of the input and a fully connected layer. The number K is calculated as $K = \log(h) - 2$, where h represents the network size. In Figure 4.1 we show the generator's network diagram.

Initial block consists of convolution layers with stride = 1 and 3x3 filter and activation function depended on network architecture. Each ResNet block consists of two convolution layers with stride = 1 and 3x3 filters then a subsampling convolution layer with stride = 2 and 3x3 filter except the last block. The number of filters is proportional the number of blocks(NoB), in our approach there exists 128xNoB filters. The coefficient multiplied by NoB comes from PG2 study. After each convolution layers, normalization and activation layers are performed respectively. The activa-

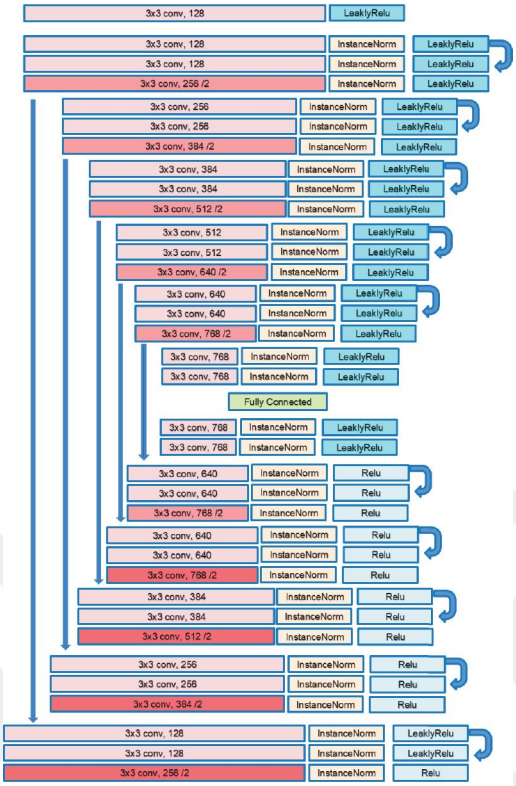


Figure 4.1: Architecture of Generator network.

tion functions used in activation layers are Leakly Relu in encoder network and Relu in decoder network. We integrate different methods in normalization layer. These methods are Instance and Batch Normalizations.

We change the depth of the DCGAN network due to the size of the image in our dataset. The modified depth of the DCGAN in our work is calculated similarly. In Figure 4.2 we show the discriminator's network diagram.

Discriminator:

We train the whole network with different parameters such as learning rate, foreground and background mask coefficient, L2 coefficient for generator and discriminator networks with Adam optimizer.

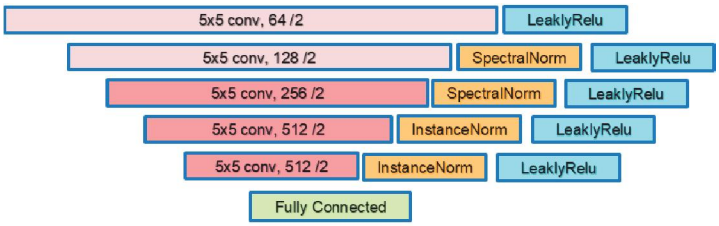


Figure 4.2: Architecture of Discriminator network.

4.2 Result

In this section, we present the performance of our proposed framework under different conditions with the usage of our comprehensive Youtube VOS and Davis dataset. The performance is analyzed in two categories with respect to qualitative and quantitative analysis. In qualitative analysis, we investigate our proposed method with the visual perception of source person, target pose and background information. We focus on analyzing the effect of different variables like input pairs, body structure, masks, learning rate, reconstruction coefficients on qualitative analysis. In addition to qualitative analysis, we perform quantitative analysis with IS, DS, SSIM and L1 to analyze the behavior of the proposed approach under different conditions for input pairs.

4.2.1 Qualitative results

In this section, we conduct our tests in 6 different categories and provide a comprehensive analysis of these categories. These are:

1. We offer extensive analysis over 3 different test sets mentioned in Section 3.3 to examine how our generator network will behave in situations that have visually similar to what they see during training and those that do not.

2. We present a comprehensive review of the scenarios where our generator network is whether successful or not in pairs with different body structures.

3. In order to examine the effects of masked training, we provide analysis results based on visual examples that compare mask and warping approaches.

4. We compare TTUR approach with the different learning rates to achieve better optimization and examine the impact of the discriminator network.

5. We examine the effects of different L2 coefficients to examine the effect of the pixel-based reconstruction loss function.

6. In order to examine the effect of our foreground and background masks, we focus on the foreground or background with different coefficients.

For images, high-frequency content, i.e. edges, is important for visual quality, and in these 6 different categories, we will focus on the details of the body's appearance, i.e. the production performance of high-frequency information, as well as other low-frequency information such as the color of clothing and general structural information determined by the target pose.

4.2.1.1 Input pair from different scenario

We divide our test scenarios into 3 different categories. The first one is a pair of images of people that exists on the test set in different poses. In this way, we are able to examine our success in pictures that we have not seen during training process. In the second category, we choose either source person or target person from the test set and the one person that is not picked from test set is assigned from training set. In this procedure, we select the pose of person on training/test set on condition that this pose is not similar with the person from the test/training set. In this way, when we select the source person from the test set, we test whether we can create the appearance of the person, or when we select the target person from the test set, we check whether we can synthesize the background. In the third category, actor and stuntman have different pose images in the training set, but we select other pose images of these people that do not exists in training set to analyze our approach. In this way, we measure our success with pictures which are not directly exist in our dataset, but similar pictures are available.

Figure 4.3 shows the results obtained for the actor and stuntman with different poses for test-test and training-training scenarios. In both categories, the 1st column shows the source person we want to produce as an appearance, the 2nd column shows the

pose and background we want to produce as the target. The 3rd column is the output of our generator model and has the appearance of the source person in the background and the pose of the target person.

In the train-train category, images of two people in different poses are included in the training set. In Figure 4.3 we observe that our approach can naturally integrate the foreground person into the corresponding background when we use foreground and background images visually similar to what the network saw during training process. In the 5th row of the Figure 4.3, our approach does not interfere the front side of the motorcycle by generating the person's right leg and does not deform the realism of the picture. In the first 3 rows, we show the image of the boxer sitting on a swing, given as a source in different poses. Since our network saw the images of the boxer in different poses during the training process then those here, the boxer is generated wtih full equipment such as gloves and headgear. However, we are having trouble generating the shoes. In the 6th and 7th rows, we show the production of a climber in different poses in the pose and background of a lady. Although the bag of the mountaineer is very large, this bag is generated in size between the bag of the mountaineer and the lady in the picture produced. This is a result of the mask and this condition will be detailed in the following sections.

We create the test-test category from couples where no images of any of the actors and stuntmen were used during the training phase. In Figure 4.3, when we use images away from the distribution of the background and foreground images of the training set, we observe that while our model produces the person realistically in the targeted pose with desired scene, we cannot achieve a similar success in the train-train category while producing the person's appearance. The training dataset has many different background scenes, and there are frames with similar backgrounds within the video. In addition, since a scenario that needs to be filled with background information due to body differences is not included in this figure, backgrounds are produced according to the scene of the target person. More specifically, our network learns to replicate the target's scene as the background is produced in people with similar body builds or heatmaps. We give examples of background production in different body structures in the following sections. Although our network does not have problems in transferring the pose on images that it has not seen before, it has a problem of not

(a) Demonstration of pairs from the test-test dataset

(b) Demonstration of pairs from the train-train dataset

Figure 4.3: Pose and target guided person image generation from pairs from the same dataset type (test-test, train-train).

producing high resolution information about the foreground. Our approach is able to generate the uniform dressing information of the person wearing a yellow coat in 6th row, but could not preserve the high frequency textural information observed in rows of 3 and 5.

Figure 4.4: Extensive demonstration of pairs from the test-test dataset.

In Figure 4.4, different results are given for the test-test category. While the person wearing a uniform, yellow color coat in the 2nd line is produced roughly, although the appearance information is not realistic, the people with contrast and color changes in their clothes on the 1st and 3rd rows cannot be synthesized because they have high-frequency information. In the 4th and 5th rows, the different behaviors of our model can be observed when the body structure of the target person is changed. We observe a better performance when the body structures of the target and source person are close to each other. In row 8, we could not create the leg of the person who was occluded by someone else.

(a) Demonstration of pairs from the test-test dataset

(b) Demonstration of pairs from the test-train dataset

Figure 4.5: Comparative representation of pairs in test-test and test-train dataset.

In category test-train, one of the actors or stuntmen is on the test set, while the other has images in different poses on the training set. In Figure 4.5a, we show the result that the person in the black t-shirt in the test-test set produced according to the pose and background of another person in the same set. In Figure 4.5b, we give the results of the same person in the black shirt with the boxer who has images in different poses in the training set. In Figure 4.5a, the person wearing a black shirt cannot be created in a realistic way, while in Figure 4.5b we can generate the person as shown in rows 1 and 2. The reason for this is that the clothes he wears are on other people in the training sets and his outfit contains image information that does not contain high frequency. In addition, the person to be produced as a target is thinner than himself. On the 3rd, 4th and 5th rows, we give the results of the images of a person in different poses in the training set according to the background and pose of a person from the test set. Due to the fact that our network uses pictures of the boxer in different poses during the training phase, our approach can produce the figure on the boxer's back.

4.2.1.2 Body structure effect in poses

In this section, we provide the results of our masked GAN training together with its success and failure due to different body structures in the test-train and train-train scenarios. In the model used during the tests, the learning rate of the discriminator is 0.0008 and it has been trained with foreground and background masks.

In Figure 4.6, we show the results of our masked GAN in the test-train scenario. In this figure, the 1st and 2nd columns respectively represent the source and target image. Column 3 contains the integration of two different masks. First mask is obtained from the given source person and this source person's pose that is similar to the pose of target person as given in eq. (3.2). Second mask is retrieved from the intersection of the background of the source person with similar pose of given target person and target person as given in eq. (3.3). More specifically, this image is acquired sequentially through the following steps. First, the foreground and background information of the source and target images are extracted. Then, to understand how much the body structures of these people overlap, the intersection of the foregrounds described in the Section 3.4.1 is found. A similar process is done in the background. Afterwards, the

foreground intersection is applied to the source image as pixel-based multiplication, and the image of the source person in the region where the source and target intersect is obtained. A similar process is done in the background. After obtaining the background masks of the persons, the background intersection of the target and source persons is calculated and then applied to the target image as pixel-based multiplication. The assemble of the resulting intersected foreground and background image is demonstrated in the 3rd column in Figure 4.6. Finally the last column shows the results obtained by our network. In addition, the people in the 1st row 1st column and 2nd row 2nd column in this figure are only in the test set, the other people have images in different poses in the training set.

In all the rows of this figure, there is a great lack of information in the foreground texture due to the non-overlap of the foreground masks in the persons given in the 3rd column. In all of the results produced in column 4, we observe that the gaps caused by the masks were filled in a meaningful way thanks to the adversarial loss function and the learning to manipulate the pixels according to the pose information using the source and target person image.

In first row of Figure 4.6, we observe that our network produces certain parts of the motorcycle to preserve the reality of the scene, while network produces the right leg to synthesize the person in a meaningful appearance. While our network significantly fills the large gaps observed in line 2, it also takes care not to spoil the appearance of the person next to it. In row 3, column 4, we observe red local information around the person. This is because while we train our network, we randomly give images of the target and source person to the discriminator network, ensuring that the generated images contain realistic backgrounds and people. However, when our network sees one of the source or target person as the majority, it updates the generator according to this information. In 3th row, we observe the effect of this situation. In 4th and 6th row, there is no information about the leg part of the source person, so we observe that our network cannot fill the leg part texturally while filling the information in the upper body in a meaningful way. While forming the body shape as expected in all results depending on the pose information, we have problems in producing the textural information of the people's appearance in the test set, which we mentioned in the Section 4.2.1.1.

Figure 4.6: Illustration of the body structure effect of the pairs in the test-train set.

In Figure 4.7, we show the results of the training with masks in the train-train scenario similar to Figure 4.6.

Figure 4.7: Illustration of the body structure effect of the pairs in the train-train set.

We examine the failure cases observed in the test-train set in Figure 4.8. For pairs in the 3rd, 5th and 7th rows in Figure 4.8, a human pose transfer is performed over the image of a person on the test set. In the first row, a certain part of the source person is blurry and the body cannot be distinguished from the motorcycle clearly. Even tough we synthesize a certain part of the body, there exists a problem in the lower body parts, especially in the legs. Therefore the outcome of this pair is not good enough. In the second row, the helmet is defined as a part of the background mask so that we see it in the last column. On the other hand, we have some partial generation in our source person but there exists some problem related to the arm again due to localization error foreground mask. In the 4th row, the blurry effect on the last column is a result of generating the source person that does not exist in the training set. Also, we see that adversarial generation become more aggressive so that the right arm at the target person exists in the background.

We use adversarial and reconstruction manner together in our training. Reconstruction type learns to establish the relationship between the given input pairs and the body resolves the gaps caused by structural differences to a certain degree on its own. The performance of the reconstruction approach decreases as the regions that need to transform in order to manipulate the pixels move away from each other. The adversarial approach not only makes the produced images more realistic but also fills the gaps caused by body differences in a meaningful way. For 5th and 6th rows in Figure 4.8, since the masks do not overlap too much, our network cannot reconstruct the image. Here, the effect of the adversarial approach comes into play, and it tries to produce the person according to the source or target person. However, since the difference between the masks is so much, meaningful textural information cannot be obtained. In addition, the poses of input pairs are different in the 5th row and the target person in this pair does not have any image during the training phase. Therefore, our network cannot extract information based on these inputs.

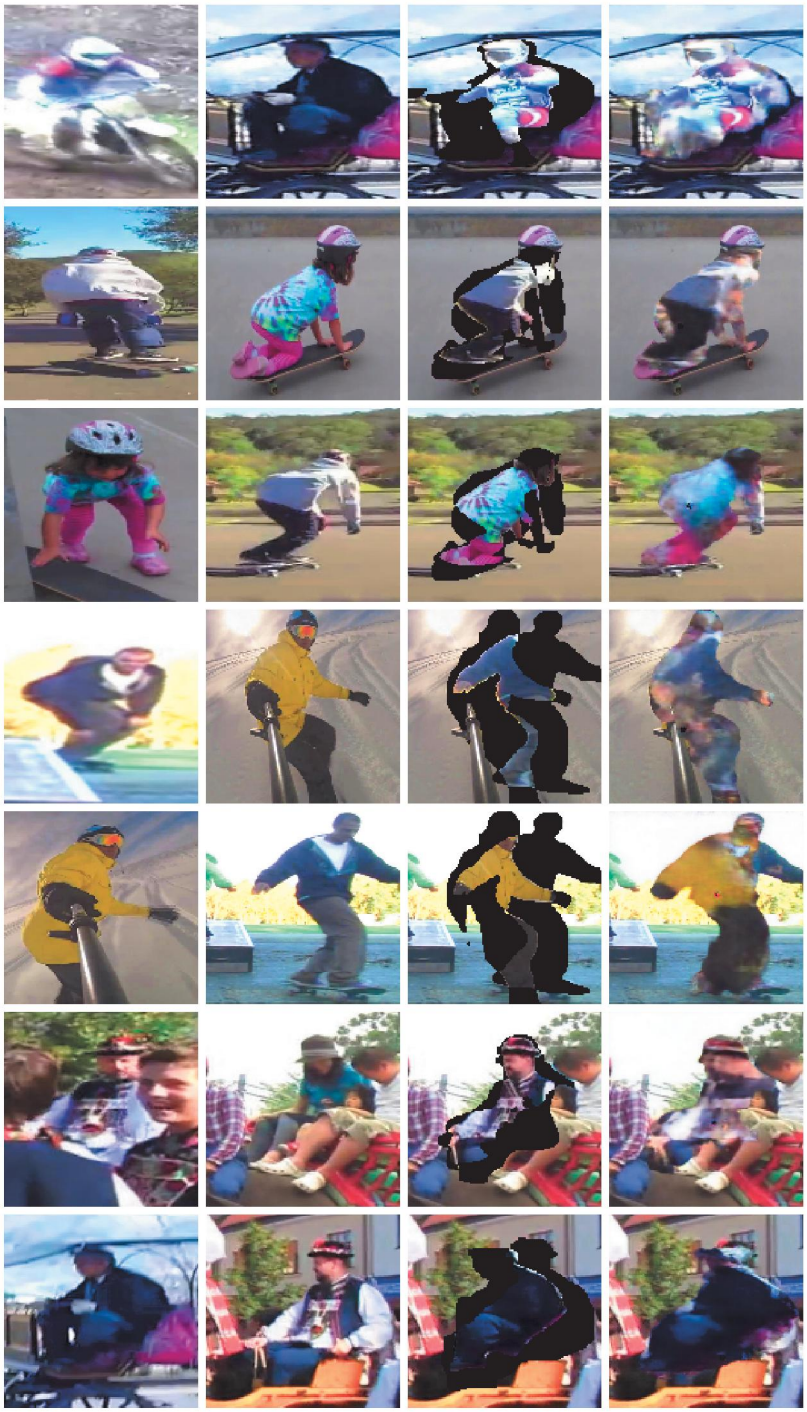


Figure 4.8: Failure cases observed for the effects of the pairs in the test-train dataset on body structure.

In Figure 4.9 we show the failure cases observed in the test-train set. In these exam-

ples, although the foreground and background masks preserve the person's appearance, these masks cannot produce the background in the desired way due to the fact that the body structures of the target and source person are very different from each other. The adversarial network arbitrarily chooses one of the images of the target and source person to fill these gaps, but cannot ideally produce a background because it does not know which image to synthesize the background.

Figure 4.9: Failure cases observed for the effects of the pairs in the train-train dataset on body structure.

4.2.1.3 Foreground and background mask effect

In this scenario, we investigate the effect of masks on person generation using our dataset. The foreground and background masks are obtained as same as Section 4.2.1.2. The loss functions in eq. (3.5.1) are used in order to understand whether our masks are useful to fill the gaps in pose generation when we have dissimilar source and target person poses. In Figure 4.10, we examine the effects of the mask and the 1st and 2nd columns respectively show the source and target person images. Column 3 contains the integration of two different masks as in the results in Section 4.2.1.2. The 4th column shows the images obtained by feeding the images in column 3 to OpenCV's Warping function. The warping function fills the gaps caused by the different foreground masks in the 3 columns with the background information. The 5th column represents the result of the warping training for reconstruction. The 6th column contains the results of the masked training.

For the warping case, it is seen that if the warped pose of given person lose a feature or information, the reconstruction of warped pose is not successful as the mask reconstructed one in general. In rows 2, 3, 10, 12 and 13, the lower body parts of generated poses fails due to this event in which we lose some information about these lower body parts with respect to column 4 of these rows. On the otherhand, in rows 1, 5 and 7 there exists some artifacts on pose generation due to the lack of intersection of body parts on foreground masks of the source and the target person for similar poses. However, mask reconstruction outperforms the warping based reconstruction in rows 8, 10, 11 and 13 when we have pairs in different body sizes. Mask reconstruction in these rows preserve the size of the source person on the otherhand the warping reconstruction basically forwards the size of the target person. Moreover, realistic generation for a body part is also demonstrated in row 13 when the leg sizes of generated person is more reasonable in the mask based reconstruction. In row 4 and 9, the mask based reconstructed person pose is effected by the source image background around boundaries due to the randomness in the fed image into the discriminator. Similar artifacts exist in warping based reconstructed person pose in row 4 and 9, but these artifacts are occured by filling the gap with the foreground mask of target person in addition to the background mask of target person.

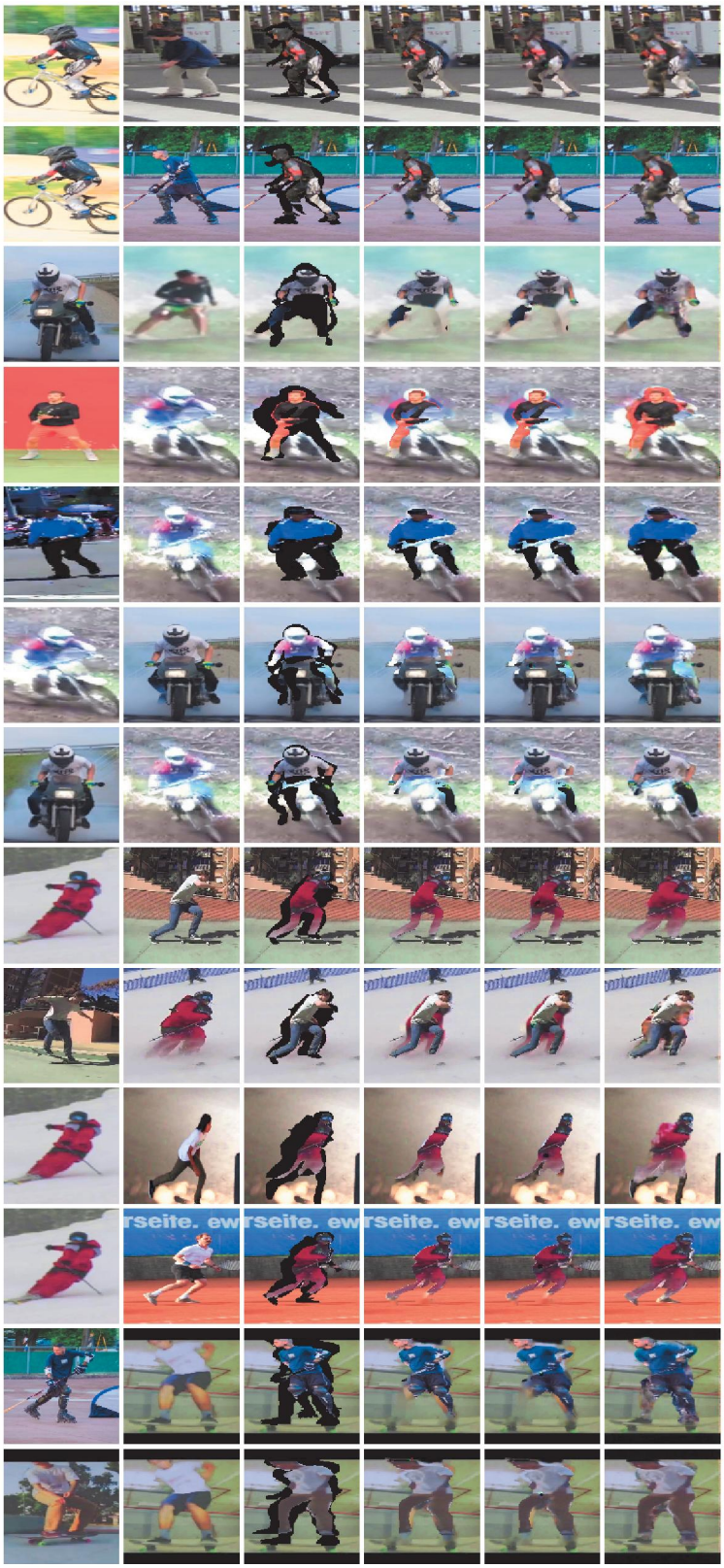


Figure 4.10: Illustration of foreground and background mask effect.

Figure 4.11: Figure showing the effects of different learning rates on discriminator behavior.

4.2.1.4 Effects of various learning rates on the Discriminator behaviour

In this scenario, we investigate the effect of different learning rates on discriminator's behaviour. If the generator learns the distribution quickly, it will become easier to fool the discriminator. This case leads discriminator to a local minima so that discriminator cannot give a feedback that does not help generator produce a wide range of transferred pose. Moreover, generator is able to produce limited poses. We analyze this scenario with the learning rates of 0.0002, 0.0004 and 0.0008.

The initial three columns are our inputs in Figure 4.11 as previous scenarios. The last three columns demonstrates the output of our approach with respect to the learning rates of 0.0002, 0.0004 and 0.0008. From Figure 4.11, it is seen that the last column is more realistic and complete than the fourth and fifth columns. As we increase the learning rates, the discriminator network pushes the generator network more to produce better results.

4.2.1.5 Effects of various reconstruction L2 coefficient on Generator behaviour

In this scenario, we investigate the effect of the Reconstruction L2 coefficient on generator behavior. Since Generator consists of two losses that are the pixel-wise mask reconstruction loss and the adversarial loss, the change in the Reconstruction

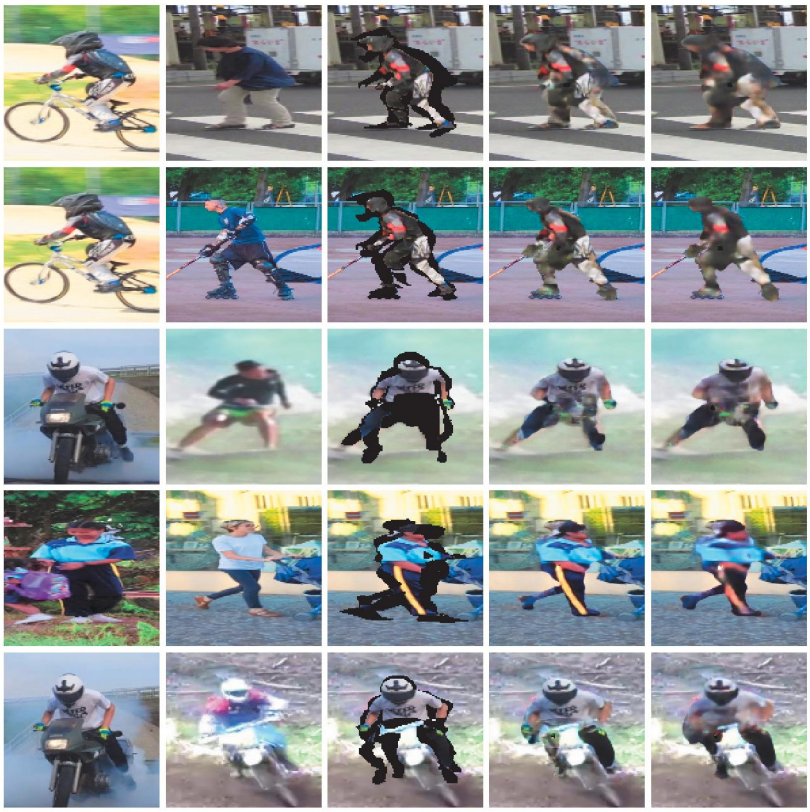


Figure 4.12: Figure showing the effects of different reconstruction L2 coefficient on generator behaviour.

L2 coefficient will be in favor of one of these losses more. We will monitor whether the adversarial loss helps our approach generate more realistic person appearance or not so that we analyze this scenario with the Reconstruction L2 coefficient of 0.01 and 0.0001.

The last two columns of Figure 4.12 represents the Reconstruction L2 coefficient of 0.01 and 0.0001. In the 4th column, the generated person appearance is more sharp and realistic than the last column since we increase the effect of the adversarial part as expected by assigning the reconstruction L2 coefficient to 0.01.

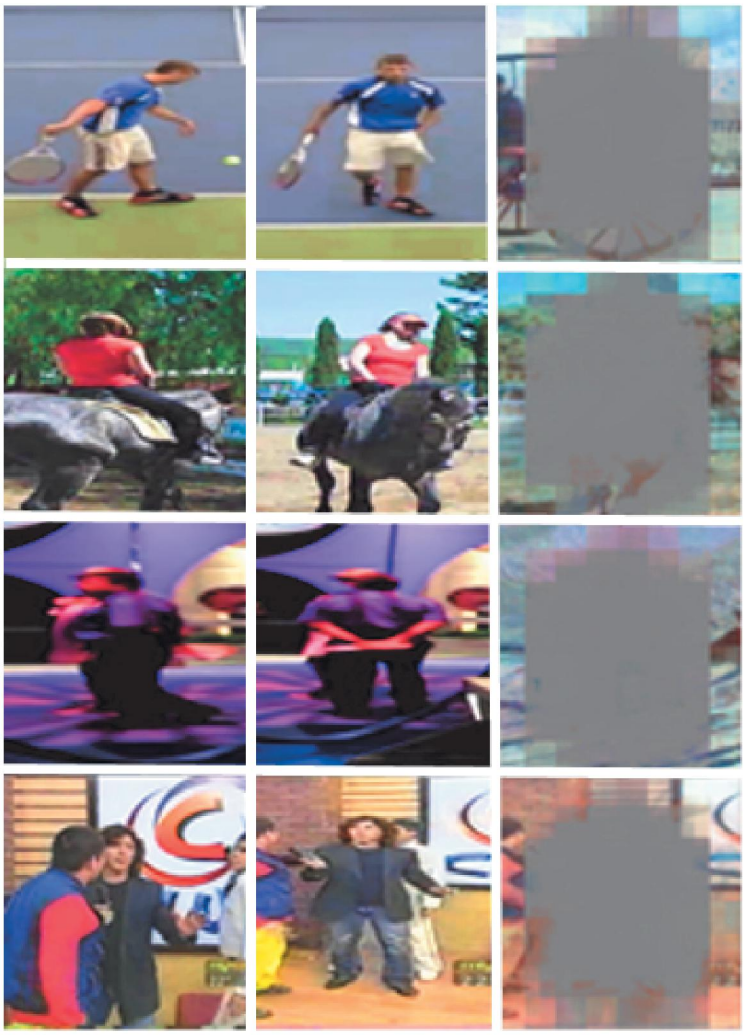


Figure 4.13: Figure showing the effects of foreground background mask coefficient on generator behaviour.

4.2.1.6 Effects of the foreground-background mask coefficients

In this scenario, we investigate the effects of the Foreground and Background Mask coefficients on generator behavior. Foreground and Background Mask coefficients decide the weight of these masks on appearance generation since we limited our generation framework to mask reconstruction. Foreground mask contains almost all information of source person so that the success of this person's pose generation with respect to target person will be effected by the change in these coefficients. We an-

alyze this scenario by changing the Foreground Mask coefficient to 1 rather than the previous coefficient of 30 that is used in the previous scenarios.

The last column in each rows of the Figure 4.13 shows that the mask reconstruction fails to generate our source person since we decrease the effect of the foreground mask.

4.2.2 Quantitative results

Although there are various types of structural or photo-realistic measurement criteria that are widely used in the literature to measure the synthesized image quality, there is no general metric that can effectively evaluate the appearance and shape consistency of these images like a human eye. Therefore, we adopt different structural or perceptual criteria such as IS, SSIM, L1, DS metrics which are widely used in the literature, and report our performance in these metrics. We use the Structural Similarity (SSIM) [96] and L1 loss function to measure the structural accuracy of our results. L1 loss and SSIM provide contextual information but do not distinguish subtle differences like a deformed body part. SSIM leans on global covariance and means of the images to estimate the structure similarity but this is insufficient to measure shape consistency.

Therefore, different metrics have been proposed to control whether photo-realistic images are produced or not. One of these metrics is Inception Scores (IS) [81]. Although many studies in the literature on Human Image Generation share IS performances, we have observed that this criterion provides limited information on in-class object production as it is based on entropy calculated on classification neurons. We do not want to define our success based on the results we derive from this metric, as our work is only on human image synthesis. Therefore, we present the performances we obtained from the Detection Scores (DS) [97] metric, which is a classification-based metric and calculates the similarity of the produced images to the person class. Detection scores correspond to the maximum confidence of score and we use the pre-train detection network which is the state-of-the-art object detector SSD [98], trained on Pascal VOC 07 for this purpose. Considering the demanding samples in the Pascal VOC 07 [99] data set, we believe that the high accuracy obtained from the SSD can

be used as an indicator of how realistic the images created are. Unlike structural metrics, IS and DS use image classifiers and object detectors to evaluate image quality and do not care about shape consistency. Since there is no metric with the advantages of both approaches, qualitative data are still important in the performance of GANs.

While the calculation in DS and IS metrics is measured directly on the generated image, in SSIM and L1 metrics, it is necessary to know both the generated image and the desired image, since how much the generated image is similar to the real image is measured. Since there is no ground-truth to calculate similarity in our dataset, we recommend the mask-SSIM and mask-IS variants for both foreground and background separately to evaluate the generated scene and human appearance. In this way, we focus on measuring the synthesis quality of a person's appearance and the quality of the scene produced. While doing this, we use the intersected foreground and background masks that we mentioned in Section 3.4.1 due to the different body structures of the target and source person. Since intersecting masks do not care about the parts where the body structure of pair people is different, we cannot measure the success of how the gaps formed here are filled by our network with similarity-based approaches. As a result, mask-based criteria can be biased in favor of our method.

Our test-test data set consists of 76 different triads belonging to 20 different people. Our test-train data set consists of 1088 different triads belonging to 26 different individuals and the train-train data set consists of 3048 triplets belonging to 40 different persons. In addition, we created the train-train set to include such triads to investigate how our network will behave in people with different body structures.

4.2.2.1 Input pair from different scenario

Table 4.1 shows the values of the structural and perceptual criteria metrics obtained on test-test, test-train, and train-train sets of our training using masks and the learning rate of the discriminator is 0.0008. *fg mask* represent the foreground mask and *bg mask* represent background mask. When we examined this table perceptually manner with IS (larger for better images) and DS (higher is better) metrics, we observed the following results, similar to our expectation: (i) The foreground and background IS scores are lower than the other sets since the human and, consequently, scene diversity

	Perceptual				Structural			
Pair Type	IS	IS fg mask	IS bg mask	DS	SSIM fg mask	SSIM bg mask	L1 fg mask	L1 bg mask
test_test	2.56	1.88	2.16	0.734	0.91	0.98	0.024	0.013
test_train	4.39	2.93	3.86	0.882	0.94	0.98	0.019	0.008
train_train	3.90	2.94	2.99	0.878	0.98	0.99	0.007	0.007

Table 4.1: Comparative IS, foreground mask IS, background mask IS, DS, foreground mask SSIM, background mask SSIM, foreground mask L1 and background mask L1 metrics result of pairs on test-test, test-train, and train-train data set. Our network, the results of which are given in this table, is trained in a masked reconstruction and the learning rate of the discriminator network is 0.0008.

in the test-test set is lower than others. (ii) Depending on the number of triplets in the test-train set, the number of different people is higher than the train-train set. To put it more clearly, the number of images of the same person in different poses in the train-train set is higher than in the test-train set. Therefore, we produce images with a similar background. This causes the background IS score to be lower than the test-train set and also causes the value of the total IS score to decrease. (iii) Human detection success is 0.73 in images with a human and background that our network has never seen, measured by DS, while it is 0.88 in the test-train data set, where the training is performed over the images of one of the actors or stuntmen in different poses. This is an indication that our deep neural network produces a better generation on images with similar distribution, as we expected. (iv) The reason why the DS score (human detection performance) is lower in the train-train data set compared to the test-train is that the train-train data set, which includes people with different body structures, is a challenging problem. The reason it is a difficult problem that our network learns the missing body parts only in an adversarial manner, due to the mask structure. (v) While the IS score depends on the distribution of the images as stated in (i) and (ii), the DS score provides a more realistic feedback mechanism on human pose transfer.

Since we do not have the desired image, we compared the generated actor's image in the stunt's pose with the image in which the actor is in the stunt's pose with the help of the foreground mask only as an appearance with $||FM \odot I_{12} - FM \odot G(I_{11}, I_{IJ}, H_{IJ})||_1$ formula. We also compared the real image of the stunt with the help of the background mask to check the similarity of the background with $||BM \odot I_{IJ} - FM \odot G(I_{11}, I_{IJ}, H_{IJ})||_1$ formula. That's why our structural metrics are divided into foreground and background. When we examined Table 4.1 structural manner with SSIM (larger for better images) and L1 (lower is better) metrics, we observed the following results, similar to our expectation: (i) Our SSIMs (network's similarity rate) with both foreground and background masks increase as go from human and background images, which have not seen any images, to images with similar distribution, as we expect. (ii) Since the human appearance contains more high-frequency content, resembling the real image is a more difficult problem. Therefore the foreground's SSIM score is lower than the background. (iii) L1 is inversely proportional to SSIM, and with this logic, we observe similar results on L1 (the higher the training set, the lower the score).

	0		1		2		3	
Pair Type	N	SSIM fg mask	N	SSIM fg mask	N	SSIM fg mask	N	SSIM fg mask
test_test	8	0.945	8	0.907	46	0.903	14	0.887
test_train	258	0.950	242	0.949	284	0.924	304	0.906
train_train	1939	0.982	655	0.981	334	0.974	120	0.963

Table 4.2: This table shows the foreground SSIM scores obtained in the different number of missing body parts/joint points (0, 1, 2, and 3) for different test categories (test-test, test-train, and train-train) as well as gives how many samples were tested in the test category for the relevant category/number of the pose. The numbers (0, 1, 2, and 3) in the top row show the number of joint poses that were occluded. N indicates how many samples are in the respective pair type (test-test, test-train) and in missing body part (0, 1, 2). Our network, the results of which are given in this table, is trained in a masked reconstruction and the learning rate of the discriminator network is 0.0008.

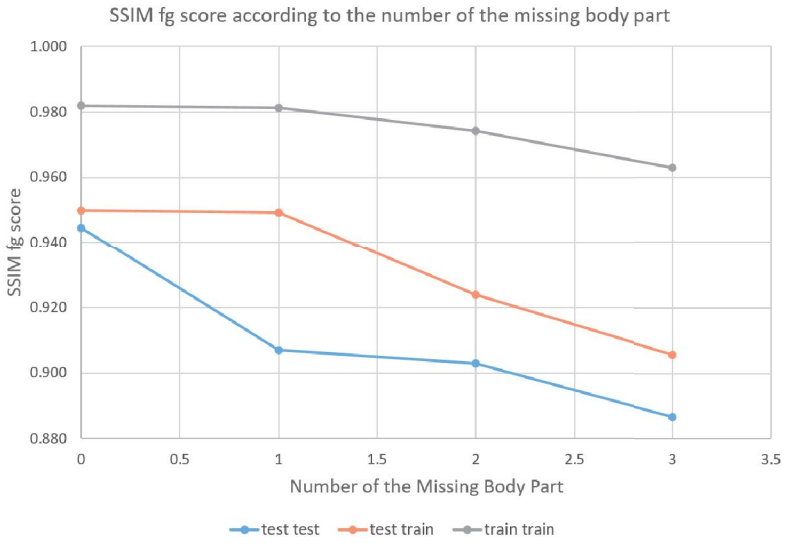


Figure 4.14: This figure shows the graph of the SSIM scores of the test pictures in different categories given in Table 4.2, depending on the number of occluded poses.

Table 4.2 shows the foreground SSIM scores obtained in different number of missing body parts and test-test, test-train, and train-train categories to measure the image generation success of the desired people who were occluded. The reason why we do not prefer IS or DS metric measurement methods, which are perceptual types, is that these metrics are based on the success of another network such as human detection, and the performance of these networks for missing body parts in the Youtube and Davis datasets is unknown. During the elimination of the images in the data set, the maximum number of missing poses is 3 because the images of people whose more than 3 poses are occluded or whose body is out of the picture are not included in the data set. The number of samples depending on the number of the missing joint poses in each test category is variable, and these variable numbers are indicated with the N columns of the table because the test category is not set to have an equal number of samples depending on the number of occluded body parts when dividing it into parts. The increase in SSIM scores observed in Table 4.1 due to the realization of training on samples with similar distribution while moving from the test-test category to the train-train category is also observed in Table 4.2.

In Figure 4.14, we present the performance graph of the foreground SSIM scores in different test categories is obtained from Table 4.2 according to the number of occluded part. This graph generally shows that the generation success decreases as the number of missing body parts increases in the desired person with the target pose. Performance in the test-test category, shown in blue, decreased with more aggressive slopes compared to the other categories according to the increasing amount of missing body parts. The reasons for this are that our network has more difficulty in generating people it has never seen, and the effect of the error on the SSIM score is higher due to the fact that the pictures in the test-test category are less than the other categories. In the test-train and train-train category, the generation success in the images of people with 1 joint point missing is very close to the generation success in the images of people without missing any body parts. In human images with missing 2 and 3 joint points, they have slopes close to each other. In the test-test category, the slope between the category of pictures with no occlusion of any pose and the category of pictures with occlusion in 1 pose is greater than the slope when going from 1 to 2. This is because the number of samples with 1 missing body part is 8, while the number of

samples without 2 pose information is 46, and a situation observed in a set with a small amount of sample (1 picture having a low SSIM score) has a greater effect on the overall distribution.

4.2.2.2 Effects of the various learning rate, coefficient and mask

In this section, we present the perceptual and structural metric results obtained on the test-test, test-training data sets of the tests, whose visual results are given in Section *Effects of the Various Learning Rates on the Discriminator Behaviour (4.2.1.4)*, Section *Reconstruction L2 Coefficient on Generator Behaviour (4.2.1.5)* and Section *Foreground and Background Mask (4.2.1.3)* in separate tables and analyze them in detail.

In Tables 4.3, 4.4 and 4.5, *Warping* is the model we train for human pose transfer over images that we interpolate without using a mask, *LR 2, LR 4 and LR 8* are the trainings we do with the learning coefficients of the discriminator network, which we give as 0.0002, 0.0004, 0.0008 by respectively, and *L2 coef* is the model we train by giving the coefficient of the generator network 0.01. In Table 4.3, we provide the Table scores of the structural and perceptual metrics of these trainings calculated on the test-test data set. Table 4.4 presents similar results for test-training and Table 4.5 for training-training set.

When we examined this Table 4.3 perceptually and structural manner, we observed the following results, similar to our expectation: (i) Since the number of different people in the test-test set is 20, the IS score does not produce meaningful results for this set. (ii) If we examine another semantic metric, DS, as we expect, the score of our masked trainings is higher than the training we do not use. (iii) We observe that performance increases when we increase the learning rate of the discriminator network. (v) The score decreases when the L2 coefficient is high, as the emphasis on the discriminator and generator networks reconstruction manner instead of adversarial causes blurry images. (vi) We observe similar effects on SSIM scores.

When we examined this Table 4.4 and Table 4.5 perceptually and structural manner, we observed the following results, (i) The situations we observed in the test-test set

	Perceptual				Structural			
Network	IS	IS fg mask	IS bg mask	DS	SSIM fg mask	SSIM bg mask	L1 fg mask	L1 bg mask
Warping	2.40	1.75	2.24	0.682	0.89	0.99	0.030	0.003
LR 2	2.47	1.65	2.08	0.691	0.90	0.973	0.025	0.022
LR 4	2.21	1.65	2.19	0.792	0.90	0.978	0.025	0.011
LR 8	2.55	1.87	2.15	0.798	0.91	0.981	0.024	0.013
L2 coef	2.24	1.71	2.16	0.784	0.90	0.973	0.026	0.015

Table 4.3: The effect of different learning rate, reconstruction L2 coefficient and mask on the **test-test** data set for human pose transfer success.

are also valid test-train and train-train set. (ii) However, unlike the test-test set, while our learning rate increases, there are a small decrease in our DS score. As can be seen in Figure 4.8, the body structures of some of our samples in the test-train set are different from each other, and since our network masks do not explicitly define the necessary transformation in cases like the 4th and 5th rows. Because the discriminator does not know scene or human information as a condition in our network, it cannot always fill large gaps in a meaningful way. Our discriminator network which decides whether image is realistic or not, depending on the picture of the actor or stuntman. A similar manner is seen in Figure 4.9 for the train-train set.

	Perceptual				Structural			
Network	IS	IS fg mask	IS bg mask	DS	SSIM fg mask	SSIM bg mask	L1 fg mask	L1 bg mask
Warping	4.339	2.878	3.990	0.865	0.922	0.998	0.024	0.003
LR 2	4.346	2.787	3.886	0.910	0.940	0.985	0.020	0.015
LR 4	4.577	2.786	3.924	0.893	0.938	0.990	0.020	0.008
LR 8	4.393	2.928	3.856	0.883	0.940	0.988	0.020	0.009
L2 coef	4.342	2.746	3.897	0.898	0.930	0.983	0.022	0.011

Table 4.4: The effect of different learning rate, reconstruction L2 coefficient and mask on the **test-train** data set for human pose transfer success.

	Perceptual				Structural			
Network	IS	IS fg mask	IS bg mask	DS	SSIM fg mask	SSIM bg mask	L1 fg mask	L1 bg mask
Warping	3.727	3.080	3.025	0.988	0.998	0.005	0.002	0.890
LR 2	3.733	2.796	3.036	0.994	0.995	0.003	0.009	0.959
LR 4	3.709	2.926	2.984	0.993	0.996	0.003	0.004	0.945
LR 8	3.907	2.939	2.996	0.980	0.993	0.007	0.007	0.891
L2 coef	3.807	2.839	2.896	0.970	0.993	0.007	0.007	0.878

Table 4.5: The effect of different learning rate, reconstruction L2 coefficient and mask on the **train-train** data set for human pose transfer success.

CHAPTER 5

CONCLUSIONS

In this thesis, we proposed a novel approach that attacks the problem of background information generation in pose guided person appearance generation. Firstly, we analyzed input pairs from different scenarios in which whether source or target person exists on given dataset or not to understand the accuracy of pose generation. Secondly, we analyzed the body structure effect on foreground and background masks to pose generation. Thirdly, we analyzed the effect of the discriminator network on the adversarial part of the generator network. Fourth, we analyzed the effect of the reconstruction process on the generator network whether the reconstruction will be in favor of adversarial or mask reconstruction. Finally, we analyzed the foreground and background mask coefficient to understand the importance of foreground mask.

The main contributions of this thesis can be summarized with 4 items. The first main contribution is that the proposed "a person's image-generating framework" can synthesize a new person image of that person in the novel target pose by preserving the background information of the target person. Second, the proposed approach consists of a novel joint mask loss to encourage the model to separate the human body view from the source image and the background view from the target image. Another main contribution is that we present a comprehensive qualitative and quantitative experimental evaluation of our approach including ablative studies. Finally, we introduce a new benchmark by adapting the video segmentation datasets which are derived from YouTube VOS and Davis to create source and target poses in different combinations.

We believe that a promising direction for future work is to improve the articulation handling of the convolutional neural network. For this purpose, mechanisms such as deformable convolutional networks can be utilized within the proposed framework.

In addition, we also believe that improvements in the dataset size and synthetic pair creation methodology are likely to improve the final prediction quality.

REFERENCES

[1] L. Ma, X. Jia, Q. Sun, B. Schiele, T. Tuytelaars, and L. Van Gool, "Pose Guided Person Image Generation," *arXiv:1705.09368 [cs]*, Jan. 2018.

[2] G. Balakrishnan, A. Zhao, A. V. Dalca, F. Durand, and J. Guttag, "Synthesizing Images of Humans in Unseen Poses," *arXiv:1804.07739 [cs]*, Apr. 2018.

[3] A. Siarohin, S. Lathuilière, E. Sangineto, and N. Sebe, "Appearance and Pose-Conditioned Human Image Generation using Deformable GANs," *arXiv:1905.00007 [cs]*, Oct. 2019.

[4] L. Ma, Q. Sun, S. Georgoulis, L. Van Gool, B. Schiele, and M. Fritz, "Disentangled Person Image Generation," *arXiv:1712.02621 [cs]*, June 2018.

[5] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," *arXiv:1505.04597 [cs]*, May 2015.

[6] E. Park, J. Yang, E. Yumer, D. Ceylan, and A. C. Berg, "Transformation-Grounded Image Generation Network for Novel 3D View Synthesis," *arXiv:1703.02921 [cs]*, Mar. 2017.

[7] X. Yan, J. Yang, E. Yumer, Y. Guo, and H. Lee, "Perspective transformer nets: Learning single-view 3d object reconstruction without 3d supervision," in *Advances in Neural Information Processing Systems* (D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, eds.), vol. 29, pp. 1696–1704, Curran Associates, Inc., 2016.

[8] T. Zhou, S. Tulsiani, W. Sun, J. Malik, and A. A. Efros, "View Synthesis by Appearance Flow," *arXiv:1605.03557 [cs]*, Feb. 2017.

[9] T. Chen, Z. Zhu, A. Shamir, S.-M. Hu, and D. Cohen-Or, "3-Sweep: extracting editable objects from a single photo," *ACM Transactions on Graphics*, vol. 32, pp. 1–10, Nov. 2013.

[10] Y. Zheng, X. Chen, M.-M. Cheng, K. Zhou, S.-M. Hu, and N. J. Mitra, "*Interactive images*: cuboid proxies for smart image manipulation," *ACM Transactions on Graphics*, vol. 31, pp. 1–11, Aug. 2012.

[11] J. Walker, K. Marino, A. Gupta, and M. Hebert, "The Pose Knows: Video Forecasting by Generating Pose Futures," *arXiv:1705.00053 [cs]*, Apr. 2017.

[12] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, G. Liu, A. Tao, J. Kautz, and B. Catanzaro, "Video-to-Video Synthesis," *arXiv:1808.06601 [cs]*, Dec. 2018.

[13] A. Siarohin, E. Sangineto, S. Lathuiliere, and N. Sebe, "Deformable GANs for Pose-based Human Image Generation," *arXiv:1801.00055 [cs]*, Apr. 2018.

[14] Z. Zhu, T. Huang, B. Shi, M. Yu, B. Wang, and X. Bai, "Progressive Pose Attention Transfer for Person Image Generation," *arXiv:1904.03349 [cs]*, May 2019.

[15] B. Zhao, X. Wu, Z.-Q. Cheng, H. Liu, Z. Jie, and J. Feng, "Multi-View Image Generation from a Single-View," *arXiv:1704.04886 [cs]*, Feb. 2018.

[16] P. Esser, E. Sutter, and B. Ommer, "A Variational U-Net for Conditional Appearance and Shape Generation," *arXiv:1804.04694 [cs]*, Apr. 2018.

[17] W. Liu, Z. Piao, J. Min, W. Luo, L. Ma, and S. Gao, "Liquid Warping GAN: A Unified Framework for Human Motion Imitation, Appearance Transfer and Novel View Synthesis," *arXiv:1909.12224 [cs, eess]*, Oct. 2019.

[18] N. Neverova, R. A. Guler, and I. Kokkinos, "Dense Pose Transfer," *arXiv:1809.01995 [cs]*, Sept. 2018.

[19] A. Pumarola, A. Agudo, A. Sanfeliu, and F. Moreno-Noguer, "Unsupervised Person Image Synthesis in Arbitrary Poses," *arXiv:1809.10280 [cs]*, Sept. 2018.

[20] C. Si, W. Wang, L. Wang, and T. Tan, "Multistage Adversarial Losses for Pose-Based Human Image Synthesis," p. 9.

[21] S. Song, W. Zhang, J. Liu, and T. Mei, "Unsupervised Person Image Generation With Semantic Parsing Transformation," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (Long Beach, CA, USA), pp. 2352–2361, IEEE, June 2019.

[22] J. Liu, B. Ni, Y. Yan, P. Zhou, S. Cheng, and J. Hu, "Pose Transferrable Person Re-Identification," p. 10.

[23] Z. Zheng, L. Zheng, and Y. Yang, "Unlabeled samples generated by gan improve the person re-identification baseline in vitro," 2017.

[24] W. Yang, W. Ouyang, X. Wang, J. Ren, H. Li, and X. Wang, "3d human pose estimation in the wild by adversarial learning," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5255–5264, 2018.

[25] X. Wang, Z. Cao, R. Wang, Z. Liu, and X. Zhu, "Improving human pose estimation with self-attention generative adversarial networks," *IEEE Access*, vol. 7, pp. 119668–119680, 2019.

[26] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," *arXiv:1312.6114 [cs, stat]*, May 2014.

[27] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Networks," *arXiv:1406.2661 [cs, stat]*, June 2014.

[28] A. Radford, L. Metz, and S. Chintala, "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks," *arXiv:1511.06434 [cs]*, Jan. 2016.

[29] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets," *arXiv:1606.03657 [cs, stat]*, June 2016.

[30] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," *arXiv:1701.07875 [cs, stat]*, Dec. 2017.

[31] C. Lassner, G. Pons-Moll, and P. V. Gehler, "A Generative Model of People in Clothing," *arXiv:1705.04098 [cs]*, July 2017.

[32] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian, "Scalable Person Re-identification: A Benchmark," in *2015 IEEE International Conference on Computer Vision (ICCV)*, (Santiago, Chile), pp. 1116–1124, IEEE, Dec. 2015.

[33] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang, "DeepFashion: Powering Robust Clothes Recognition and Retrieval with Rich Annotations," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (Las Vegas, NV, USA), pp. 1096–1104, IEEE, June 2016.

[34] D. Liang, R. Wang, X. Tian, and C. Zou, "PCGAN: Partition-Controlled Human Image Generation," *arXiv:1811.09928 [cs]*, Nov. 2018.

[35] M. Liu, K. Wang, J. Ji, and S. S. Ge, "Person image generation with semantic attention network for person re-identification," *arXiv:2008.07884 [cs]*, Aug. 2020.

[36] T.-C. Wang, M.-Y. Liu, A. Tao, G. Liu, J. Kautz, and B. Catanzaro, "Few-shot Video-to-Video Synthesis," *arXiv:1910.12713 [cs]*, Oct. 2019.

[37] O. Gafni, O. Ashual, and L. Wolf, "Single-Shot Freestyle Dance Reenactment," *arXiv:2012.01158 [cs]*, Dec. 2020.

[38] Y. Zhou, Z. Wang, C. Fang, T. Bui, and T. L. Berg, "Dance Dance Generation: Motion Transfer for Internet Videos," *arXiv:1904.00129 [cs]*, Mar. 2019.

[39] A. v. d. Oord, N. Kalchbrenner, and K. Kavukcuoglu, "Pixel Recurrent Neural Networks," *arXiv:1601.06759 [cs]*, Aug. 2016.

[40] A. v. d. Oord, N. Kalchbrenner, O. Vinyals, L. Espeholt, A. Graves, and K. Kavukcuoglu, "Conditional Image Generation with PixelCNN Decoders," *arXiv:1606.05328 [cs]*, June 2016. arXiv: 1606.05328.

[41] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th international conference on Machine learning - ICML '08*, (Helsinki, Finland), pp. 1096–1103, ACM Press, 2008.

[42] R. Socher, E. H. Huang, J. Pennin, C. D. Manning, and A. Y. Ng, "Dynamic Pooling and Unfolding Recursive Autoencoders for Paraphrase Detection," p. 9.

[43] Y. LeCun and C. Cortes, "MNIST handwritten digit database," 2010.

[44] A. Krizhevsky, "Learning Multiple Layers of Features from Tiny Images," p. 60.

[45] E. Denton, S. Chintala, A. Szlam, and R. Fergus, "Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks," *arXiv:1506.05751 [cs]*, June 2015.

[46] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks," *arXiv:1703.10593 [cs]*, Aug. 2020.

[47] Z. Yi, H. Zhang, P. Tan, and M. Gong, "DualGAN: Unsupervised Dual Learning for Image-to-Image Translation," *arXiv:1704.02510 [cs]*, Oct. 2018.

[48] M.-Y. Liu, T. Breuel, and J. Kautz, "Unsupervised Image-to-Image Translation Networks," *arXiv:1703.00848 [cs]*, July 2018.

[49] M. Mirza and S. Osindero, "Conditional Generative Adversarial Nets," *arXiv:1411.1784 [cs, stat]*, Nov. 2014.

[50] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation," *arXiv:1711.09020 [cs]*, Sept. 2018.

[51] N. Zheng, X. Song, Z. Chen, L. Hu, D. Cao, and L. Nie, "Virtually Trying on New Clothing with Arbitrary Poses," in *Proceedings of the 27th ACM International Conference on Multimedia*, (Nice France), pp. 266–274, ACM, Oct. 2019.

[52] A. H. Raffiee and M. Sollami, "GarmentGAN: Photo-realistic Adversarial Fashion Transfer," *arXiv:2003.01894 [cs]*, Mar. 2020.

[53] S. Murali, M. R. Rajati, and S. Suryadevara, "Image generation and style transfer using conditional generative adversarial networks," in *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, pp. 1415–1419, 2019.

[54] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs," *arXiv:1711.11585 [cs]*, Aug. 2018.

[55] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-Image Translation with Conditional Adversarial Networks," *arXiv:1611.07004 [cs]*, Nov. 2018.

[56] X. Huang, Y. Li, O. Poursaeed, J. Hopcroft, and S. Belongie, "Stacked Generative Adversarial Networks," *arXiv:1612.04357 [cs, stat]*, Apr. 2017.

[57] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual Losses for Real-Time Style Transfer and Super-Resolution," *arXiv:1603.08155 [cs]*, Mar. 2016.

[58] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network," *arXiv:1609.04802 [cs, stat]*, May 2017.

[59] A. Odena, C. Olah, and J. Shlens, "Conditional Image Synthesis With Auxiliary Classifier GANs," *arXiv:1610.09585 [cs, stat]*, July 2017.

[60] Z. Wang, Q. She, and T. E. Ward, "Generative Adversarial Networks in Computer Vision: A Survey and Taxonomy," *arXiv:1906.01529 [cs]*, Dec. 2020.

[61] K. Kurach, M. Lucic, X. Zhai, M. Michalski, and S. Gelly, "A Large-Scale Study on Regularization and Normalization in GANs," *arXiv:1807.04720 [cs, stat]*, May 2019.

[62] J. Donahue, P. Krähenbühl, and T. Darrell, "Adversarial Feature Learning," *arXiv:1605.09782 [cs, stat]*, Apr. 2017.

[63] D. Berthelot, T. Schumm, and L. Metz, "BEGAN: Boundary Equilibrium Generative Adversarial Networks," *arXiv:1703.10717 [cs, stat]*, May 2017.

[64] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive Growing of GANs for Improved Quality, Stability, and Variation," *arXiv:1710.10196 [cs, stat]*, Feb. 2018.

[65] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-Attention Generative Adversarial Networks," *arXiv:1805.08318 [cs, stat]*, June 2019.

[66] A. Brock, J. Donahue, and K. Simonyan, "Large Scale GAN Training for High Fidelity Natural Image Synthesis," *arXiv:1809.11096 [cs, stat]*, Feb. 2019.

[67] T. Kaneko, Y. Ushiku, and T. Harada, "Label-Noise Robust Generative Adversarial Networks," *arXiv:1811.11165 [cs, stat]*, May 2019.

[68] X. Gong, S. Chang, Y. Jiang, and Z. Wang, "AutoGAN: Neural Architecture Search for Generative Adversarial Networks," *arXiv:1908.03835 [cs, eess]*, Aug. 2019.

[69] A. Karnewar and O. Wang, "MSG-GAN: Multi-Scale Gradient GAN for Stable Image Synthesis," *arXiv:1903.06048 [cs, stat]*, Nov. 2019. arXiv: 1903.06048.

[70] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *arXiv:1512.03385 [cs]*, Dec. 2015.

[71] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley, "Least Squares Generative Adversarial Networks," *arXiv:1611.04076 [cs]*, Apr. 2017.

[72] J. Zhao, M. Mathieu, and Y. LeCun, "Energy-based Generative Adversarial Network," *arXiv:1609.03126 [cs, stat]*, Mar. 2017.

[73] L. Metz, B. Poole, D. Pfau, and J. Sohl-Dickstein, "Unrolled Generative Adversarial Networks," *arXiv:1611.02163 [cs, stat]*, May 2017.

[74] G.-J. Qi, "Loss-Sensitive Generative Adversarial Networks on Lipschitz Densities," *arXiv:1701.06264 [cs]*, Mar. 2018.

[75] J. H. Lim and J. C. Ye, "Geometric GAN," *arXiv:1705.02894 [cond-mat, stat]*, May 2017.

[76] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved Training of Wasserstein GANs," *arXiv:1704.00028 [cs, stat]*, Dec. 2017.

[77] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral Normalization for Generative Adversarial Networks," *arXiv:1802.05957 [cs, stat]*, Feb. 2018.

[78] M. Lucic, K. Kurach, M. Michalski, S. Gelly, and O. Bousquet, "Are GANs Created Equal? A Large-Scale Study," p. 10.

[79] L. Mescheder, A. Geiger, and S. Nowozin, "Which Training Methods for GANs do actually Converge?," *arXiv:1801.04406 [cs]*, July 2018.

[80] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium," *arXiv:1706.08500 [cs, stat]*, Jan. 2018.

[81] T. Salimans, I. J. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," *CoRR*, vol. abs/1606.03498, 2016.

[82] H. Dong, X. Liang, K. Gong, H. Lai, J. Zhu, and J. Yin, "Soft-Gated Warping-GAN for Pose-Guided Person Image Synthesis," *arXiv:1810.11610 [cs]*, Jan. 2019.

[83] A. Raj, P. Sangkloy, H. Chang, J. Hays, D. Ceylan, and J. Lu, "SwapNet: Image Based Garment Transfer," p. 17.

[84] R. Yu, X. Wang, and X. Xie, "VTNFP: An Image-Based Virtual Try-On Network With Body and Clothing Feature Preservation," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, (Seoul, Korea (South)), pp. 10510–10519, IEEE, Oct. 2019.

[85] X. XIAO, P. KUANG, X. I. GU, and M. HE, "Pedestrian image generation with target pose based on the improved cyclegan," in *2019 16th International Computer Conference on Wavelet Active Media Technology and Information Processing*, pp. 174–177, 2019.

[86] W. Sun, J. H. Bappy, S. Yang, Y. Xu, T. Wu, and H. Zhou, "Pose Guided Fashion Image Synthesis Using Deep Generative Model," *arXiv:1906.07251 [cs]*, Sept. 2019.

[87] B. AlBahar and J.-B. Huang, "Guided Image-to-Image Translation with Bi-Directional Feature Transformation," *arXiv:1910.11328 [cs]*, Oct. 2019.

[88] C. Chan, S. Ginosar, T. Zhou, and A. A. Efros, "Everybody Dance Now," *arXiv:1808.07371 [cs]*, Aug. 2019.

[89] H. Cai, C. Bai, Y.-W. Tai, and C.-K. Tang, "Deep Video Generation, Prediction and Completion of Human Action Sequences," *arXiv:1711.08682 [cs, stat]*, Dec. 2017.

[90] X. Chen, J. Song, and O. Hilliges, “Unpaired Pose Guided Human Image Generation,” p. 10.

[91] S. Huang, H. Xiong, Z.-Q. Cheng, Q. Wang, X. Zhou, B. Wen, J. Huan, and D. Dou, “Generating Person Images with Appearance-aware Pose Stylizer,” *arXiv:2007.09077 [cs, eess]*, July 2020.

[92] L. Yang, P. Wang, X. Zhang, S. Wang, Z. Gao, P. Ren, X. Xie, S. Ma, and W. Gao, “Region-adaptive Texture Enhancement for Detailed Person Image Synthesis,” *arXiv:2005.12486 [cs]*, May 2020.

[93] N. Xu, L. Yang, Y. Fan, D. Yue, Y. Liang, J. Yang, and T. Huang, “YouTube-VOS: A Large-Scale Video Object Segmentation Benchmark,” *arXiv:1809.03327 [cs]*, Sept. 2018.

[94] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung, “A Benchmark Dataset and Evaluation Methodology for Video Object Segmentation,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (Las Vegas, NV), pp. 724–732, IEEE, June 2016.

[95] K. Sun, B. Xiao, D. Liu, and J. Wang, “Deep high-resolution representation learning for human pose estimation,” 2019.

[96] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, “Image quality assessment: From error visibility to structural similarity,” *Image Processing, IEEE Transactions on*, vol. 13, pp. 600 – 612, 05 2004.

[97] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, “Faceforensics: A large-scale video dataset for forgery detection in human faces,” *CoRR*, vol. abs/1803.09179, 2018.

[98] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, “Ssd: Single shot multibox detector,” *Lecture Notes in Computer Science*, p. 21–37, 2016.

[99] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results.” http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.