Simen Holmestad

Towards Consistent Full-Body Anonymization

Master's thesis in Computer Science
Supervisor: Frank Lindseth
Co-supervisor: Håkon Hukkelås
June 2022

Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

NTNU
Norwegian University of
Science and Technology

Simen Holmestad

Towards Consistent Full-Body Anonymization

Master's thesis in Computer Science
Supervisor: Frank Lindseth
Co-supervisor: Håkon Hukkelås
June 2022

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

NTNU
Norwegian University of
Science and Technology

Abstract

For applications requiring large amounts of image data, the availability and usability of such images often conflict with privacy regulations like GDPR. Naive anonymization methods (*e.g*. masking, blurring) result in images wildly different from the original ones, making the data unusable for applications requiring realistic image content. Current realistic full-body anonymization works well on single images but tends to create significant changes in the appearance of the generated person for small changes in backgrounds and poses. In this thesis, we propose a model that generates more consistent anonymizations for changes in the input, paving the way for realistic video anonymization. Our model is trained to reconstruct the original person in the image, such that the model is forced to use a latent variable $\vec{z}$ when synthesizing the person's appearance. When used to anonymize the Market1501 dataset, our model outperforms our StyleGAN baseline by a wide margin in terms of retaining the same identity across various images. The improved model has no increase in inference time, but increasing consistency leads to a decline in image quality and diversity. We will, through this thesis, discuss several challenges related to consistent full-body anonymization and propose new directions for further research in the field.

Sammendrag

Det finnes mange datasystemer som krever store mengder bildedata, men tilgjengeligheten og kvaliteten på denne bildedataen begrenses ofte av personvenrnsreguleringer som GDPR. Et problem er at banale anonymiseringsteknikker som blurring og maskering lager bilder som ser veldig annerledes ut enn originalbildene. Dette fører til at datasett anonymisert med disse teknikkene er uegnet i tilfeller der det kreves realistisk bildeinnhold. Det finnes i dag fullkroppsanonymiseringsmodeller som fungerer bra til å generere realistiske enkeltbilder, men disse skaper store endringer i en persons utseende ved små endringer i bakgrunns- og positurinformasjon. I denne rapporten legger vi fram en realistisk fullkroppsanonymiseringsmodell som gir mer konsekvente anonymiseringer ved endringer i bakgrunn og positur, og kommer dermed et steg videre på veien mot realistisk videoanonymisering. Modellen vår er trent til å rekonstruere den originale personen i bildet, slik at modellen tvinges til å bruke den latente variabelen $\vec{z}$ når den bestemmer utseendet til personen som skal genereres. Når modellen vår brukes til å anonymisere Market1501-datasettet er den overlegent bedre enn utganspunktmodellen vår basert på StyleGAN i å generere samme person på tvers av bilder. Den forbedrede modellen gir ingen økning i hvor lang tid anonymiseringen tar, men å øke konsekventheten fører til en nedgang i bildekvalitet og mangfold i de generte bildene. Vi vil gjennom rapporten diskutere flere utfordringer relatert til konsekvent fullkroppsanonymisering og komme med forslag til hva som burde unsersøkes i videre forskning på feltet.

Preface

The following pages contain my master thesis regarding how to make realistic anonymization of humans more consistent. This thesis is the final part of a 5-year master's degree in computer science with a specialization in artificial intelligence at the Norwegian University of Science and Technology (NTNU). My supervisors have been Professor Frank Lindseth and Håkon Hukkelås from the Department of Computer Science.

I would like to thank Frank and Håkon for good cooperation throughout the thesis period. Your guidance and all the discussions we have had has been of great help for moving through the fast-moving and exciting field of computer vision. I would also like to thank all the friends i have met on my journey through 18 years of education for making my life better, as well as my family for always being supportive.

Simen Holmestad

Trondheim, 13th June 2022

Contents

Figures

Tables

Acronyms

AdaIN Adaptive Instance Normalization. 21, 22

CelebA CelebFaces Attributes Dataset. xvi, 34, 35

CIAGAN Conditional Identity Anonymization Generative Adversarial Networks. xvi, 31, 32

CLIP Contrastive Language-Image Pretraining. 28, 52

CMC Cumulated Matching Characteristics. 8

COCO Common Objects in Context. 6, 13, 35, 43

CSE Continuous Surface Embedding. xvi, 13, 32, 33, 35, 36, 42–46, 51, 64, 66, 67, 71–73, 77, 78, 87

FDH Flickr Diverse Humans. xvii, 4, 36, 41–45, 54, 73, 76, 77, 87–89, 91

FFHQ Flickr-Faces-HQ. xvi, 34, 35, 38

FID Fréchet Inception Distance. xvi, 27, 28, 52, 69

GAN Generative Adversarial Networks. 5, 16–19, 21–24, 26, 31, 34, 45, 49, 69, 70, 77

GDPR General Data Protection Regulation. iii, v, 1

IoU Intersection Over Union. 7

KL Kullback-Leibler. 48, 49, 67–69, 71

LPIPS Learned Perceptual Image Patch Similarity. xvi, 29, 52, 53, 55

mAP Mean Average Precision. 6, 8

MOTS Multi-object tracking and segmentation. 76

PCA Principal Component Analysis. 12

PII Personal Identifiable Information. 1

PPL Perceptual Path Length. 77

SMPL Skinned Multi-Person Linear. xv, 11–14, 37, 77, 78

YFCC100M Yahoo Flickr Creative Commons 100 Million Dataset. 43, 76

Chapter 1

Introduction

Modern computer vision models require large amounts of labeled training images to function properly. However, gathering images of people often conflicts with privacy regulations. Systems exist for both realistic face and full-body anonymization, but no system can consistently anonymize the same body for varying poses and backgrounds. This master thesis explores how to create realistic full-body anonymization models robust to changes in pose and background, paving the way for better video anonymization.

1.1 Motivation

Computer vision datasets should preferably come from real and varied environments and contain clear, identifiable objects. However, capturing high-quality image data from self-driving vehicles poses a problem, as many images will contain identifiable persons. Strict privacy regulations (*e.g*. GDPR) make it illegal in many regions to collect and store images containing Personal Identifiable Information (PII) without explicit consent. In the self-driving vehicle case, asking every person captured on camera for consent is not feasible, so an anonymization pipeline is needed.

Pixelation, blurring, or masking of the persons in the dataset are standard anonymization techniques used in many applications, such as Google Street View [1]. However, these approaches severely distort the original data. An effective measure to counter this is to replace the depicted individuals with realistically generated people. The goal of such an anonymization pipeline is to keep the original dataset distribution intact while at the same time preserving privacy. More work on systems for realistic anonymization might open up the possibility for a larger amount of high-quality image and video datasets to be open to the public.

Several systems exist for realistic anonymization of faces. However, only operating on faces is insufficient in many cases as a person could be identified through other identifiers on their body. Realistic anonymization operating on full bodies is still in its infancy today, and the current state-of-the-art generates highly inconsistent anonymization results for videos.

1.2 Goal and Research Questions

The goal of the thesis is as follows:

- **Goal**: *Create a two-stage realistic full-body anonymization pipeline able to generate the same person consistently for various poses and image contexts.*

A more consistent realistic full-body anonymization pipeline will allow for new anonymization scenarios, such as better anonymization of videos. This, in turn, means that anonymized datasets can be better suited to tasks such as tracking, which is crucial for applications like self-driving vehicles. Our anonymization pipeline will consist of two stages where the first stage consists of finding the persons in the images, and the second stage consists of replacing these persons. To be able to reach the goal of creating such a pipeline, we will, throughout the report, try to answer several research questions (RQs):

- **RQ1**: *What are the main challenges of consistent full-body anonymization?*

The task of consistent full-body anonymization is not explored in current literature, so highlighting and searching for challenges in the domain is key to bringing the field forward. Challenges can be found by examining existing literature for similar tasks and through experimentation.

- **RQ2**: *What datasets and pose estimation methods are suited for the task of consistent full-body anonymization?*

For any computer vision task, the properties of the training dataset will severely impact how the final model will function. In our case, we have two major decisions regarding the choice of dataset. First, what kind of training images should we use for the best possible anonymizations? And second, how should the persons in the dataset be represented (*e.g.* in terms of pose information) before being sent into our anonymization model?

- **RQ3**: *How can we improve on existing anonymization techniques to make them more consistent?*

Creating a new system entirely from scratch is probably not the best way to go forward with this task. Basing the system on existing techniques and trying to find improvements for those techniques will make development swifter and possibly bring up interesting new questions.

- **RQ4**: *How can we evaluate anonymization consistency?*

Most subfields of deep learning are based on experiments, so having quantitative metrics able to describe the performance of a given system reliably is crucial. In our case, finding out how to evaluate consistency would be very beneficial for further development of consistent anonymization systems.

1.3 Research Method

We will answer the research questions through analysis of existing literature and experiments. The results from the experiments will be analyzed quantitatively by using several metrics and qualitatively through images and video. Qualitative evaluation is especially important for generative models, as creating reliable metrics assessing the quality of generated images is a difficult problem. We will analyze both the final model and several ablation models created to assess the impact of improvements made to the system throughout the thesis period.

1.4 Contributions

We propose a full-body anonymization model that generates more consistent anonymizations than previous methods. The model is trained to generate an appearance vector from the original person pixels and use this appearance vector when generating the anonymized person. By forcing the model to use this appearance vector, the model becomes less sensitive to the background and pose of the original person when anonymizing. The new model provides increased temporal consistency for video without increasing the inference time.

Our contributions can be summarized as follows:

1. We explore the task of consistent full-body anonymization, relates the task to existing literature, and highlight key challenges to overcome.
2. We create a model able to do more consistent anonymization than current state-of-the-art in full-body anonymization, paving the way for better video anonymization.

3. We discuss key challenges for our model, dataset, pose detection, and metrics and encourage further research and development in several areas.

All code used in the thesis, including config files, is included along with the thesis document to ensure reproducibility of the results. In addition, we have included final model weights for the StyleGAN baseline model and our final model. The dataset will be published as open-source at a later point in time.

1.5 Thesis Structure

Chapter 2 will introduce the reader to the field of computer vision, methods for gathering human pose information from images, generative models, and common metrics in the field. Then, chapter 3 will compare our task of consistent full-body anonymization to similar tasks, as well as examine and compare common datasets used in related research. Chapter 4 will introduce the FDH dataset, our baseline model based on StyleGAN, and improvements done to this model to make the anonymization process more consistent. In chapter 5 we will present the results of the model, both quantitatively and qualitatively, and in chapter 6 we will discuss the results of chapter 5 in terms of the research questions. Finally, chapter 7 will conclude the thesis and propose several directions for further research.

Chapter 2

Background

The field of computer vision, and particularly the parts related to generative models, has seen considerable improvements in the last decade. This chapter will first introduce the reader to the field of computer vision, common computer vision tasks, and common model architectures. We then move on to explain various computer vision models able to give information about the whereabouts of humans in an image, as these models will later be used in dataset generation. Finally, we move on to the field of generative models and GANs in particular, as GANs are currently state-of-the-art for generating realistic images and will be used in our models in chapter 4. At the end of the chapter, metrics used later in the thesis for assessing the quality of generated images will be covered.

This introductory chapter assumes the reader is familiar with basic machine learning theory and neural networks. We refer the reader to the books "Deep Learning" [2], "Pattern Recognition and Machine Learning" [3] and "Artificial Intelligence: A Modern Approach" [4] for a good introduction to the theme.

2.1 Computer Vision

From an engineering point of view, computer vision aims to build autonomous systems that could perform some of the tasks the human visual system can perform [5]. With improvements in machine learning, the introduction of convolutional neural networks, and major improvements in computer hardware, the capabilities and performance of computer vision systems have skyrocketed in the last years.

The field of computer vision is mainly driven by empirical research with a lot of experiments. Researchers compete on standardized tasks using standardized image datasets and standardized metrics for assessing the results. The field moves forward by the creation of new tasks, datasets, and metrics, as well as competition to beat the computer vision model currently having the highest metric scores for the various tasks. The current best model for a given task is often referred to as “state-of-the-art”.

2.1.1 Common Tasks in Computer Vision

As mentioned, the field of computer vision moves forward by new *tasks*, new *datasets*, new *metrics* and new *models*. As the field advances, the tasks get more challenging, the datasets get larger and more varied, the metrics become more precise and useful, and the models get bigger, more complex, or even completely changed. Below are some of the most common computer vision tasks with much recent research.

Classification

Classification is the task of assigning a label to an image given a limited set of labels. A classification task could be to distinguish images of cats and dogs or to predict handwritten digits from images. The most influential benchmark dataset for image classification is the ImageNet Dataset [6], where the benchmarking version of the dataset contains 1000 different classes. The performance of classification models is often measured through the *accuracy*, meaning the amount of class predictions it got right, or *top-k accuracy* for different k. Top-k accuracy metric means the amount of predictions where the right class was present among the k most probable class predictions from the modl.

Object Detection

The goal of object detection is to predict the location and class of multiple objects in a single image. Object detection is more general than classification, as the image is not limited to having just one type of object present. In addition, the number of objects present and their position can be read from the results. Object detection is based on bounding boxes in image coordinates, so to create object detection training data, one must create a bounding box with a class label for each object present in the image. Object detection is useful in many scenarios, such as detecting people from camera data in a self-driving vehicle. A common benchmarking dataset for object detection is the Common Objects in Context (COCO) Dataset [7], containing 80 classes for object detection. The most common metric for object detection models is Mean Average Precision (mAP), which takes both precision and recall of the model into account. Precision in this case is the amount of given predictions being correct, while recall is the amount of total objects the model was able to find.

Segmentation

Segmentation is the task of predicting class labels for individual pixels, thus segmenting the image into different regions. This allows for more fine-grained positioning than object detection, as one is not limited to bounding boxes. There are multiple segmentation tasks, such as semantic and instance segmentation. Semantic segmentation gives every pixel in the image a class label, while instance segmentation takes it one step further by also separating instances of the same class from each other. Instance segmentation can, in this sense, be seen as a combination of segmentation and object detection as it gives both a class label and a mask for every object present. It is worth noting that when moving from classification to object detection to segmentation, the labeling process becomes increasingly more time-consuming as more fine-grained information is needed. A common metric used for segmentation is Mean Intersection Over Union (IoU), meaning the mean mask intersection over mask union for every class.

Object Tracking

Object tracking extends object detection to videos, where detected objects are tracked across frames. Each object has an associated id, and the model must ensure that the ids are consistent throughout the video. Common problems in object tracking include losing an object for some time and assigning a new id to it, as well as id shifts when two objects come close to each other.

Keypoint Estimation

Keypoint estimation is the task of predicting points of interest related to an object in an image. These points can, for example, correspond to human joints or edges of clothes [8]. Both Single(fixed)-Object Keypoint estimation and Multi-Object Keypoint estimation exist, with Multi-Object Keypoint estimation being significantly more challenging, as it is not known beforehand how many objects will be present in each image. Possible use-cases of keypoint estimation include analyzing videoes of moving athletes or predicting certain diseases based on footage of how babies move.

Person Re-Identification

Person re-identification is the task of matching persons of the same identity in images. The images in question can be captured with various angles, viewpoints, lenses, and cameras, and the people in the images can be captured at different points in time. Given a probe image (query), the person re-identification task is to search in a gallery (database) for images that contain the same person. [9]. To assess the quality of the system, a Cumulated Matching Characteristics (CMC) curve is often used, showing the probability that a query identity appears in different-sized candidate lists. However, when multiple ground truth images exist, the amount of matched candidates is not taken into account with CMC, making mAP a more accurate metric [9].

2.2 Extracting Image Features with Convolutions

A *convolution* is a mathematical operation taking two functions as input and producing a new function. In computer vision, the word convolution is a misnomer, as the actual process being done is often a *cross-correlation*, defined in the discrete 2D case as:

$$F[i,j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} h[u,v] \cdot I[i+u, j+v] \tag{2.1}$$

Here, F is the output function while h and I are input functions. Both F, h, and I are 2-dimensional discrete functions that are used to represent greyscale images, while k is half the *width* of h rounded down. For most computer vision purposes, the cross-correlation is done with an input image F and a very small image h (typically 3x3), known as a *kernel*. Often, we say that a kernel is looking for *features* in an image, so the resulting grid output from a convolution is often called a *feature map*. One example of a hand-crafted kernel is the Sobel kernel [10] seen in Equation 2.2. The Sobel kernel is designed to check for vertical edges in an image, and an example of its use can be seen in Figure 2.1. While the kernel in Equation 2.2 is made for greyscale images with one input channel, a general image kernel will have as many channels as the input image (3 for an RGB color image). In most computer vision scenarios today, the values of a kernel h for convolution are not hand-crafted but rather optimized through some machine learning training process.

$$S_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \tag{2.2}$$

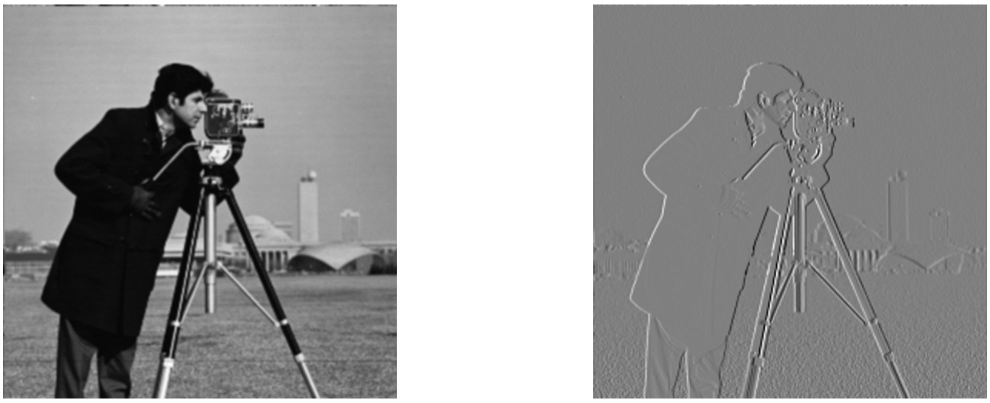


Figure 2.1: Edge detection by computing a cross-correlation with the horizontal sobel kernel S_x from Equation 2.2. The kernel is passed through the image pixel by pixel and the output pixel is calculated by a weighted summation of the 3x3 pixel grid surrounding the input pixel. **Left**: The original greyscale image, **Right**: The resulting image from applying the kernel, often referred to as a *feature map*.

A convolutional neural network will typically contain many *convolutional layers*, each with its own set of kernels looking for features. Each layer will apply its kernels to the input image (or set of feature maps) and stack the output feature maps channel-wise to create the input for the next layer. In addition, a non-linear *activation function* is applied to the values of each feature map to ensure the network can learn non-linear mappings. A typical convolutional architecture consists of multiple convolutional layers combined with downsampling operations (such as max-pooling) which cuts the spatial resolution of the input in half. An example of such a convolutional architecture used for classification can be seen in Figure 2.2. Convolutional neural networks work better than fully-connected networks on images because they are translational invariant and use the same parameters across the whole image. This makes them more robust to small changes in the input, which in turn makes them better at generalizing to new, unseen images.

When a model such as the one in Figure 2.2 is trained, it is expected that the feature maps, as they get smaller and smaller, will contain more semantic meaning. The first feature maps might contain information regarding edges, while the next might contain more information about textures and patterns. In the end, the activations in the feature maps might represent more semantically meaningful information, such as the presence of a dog. The set of all convolutional layers and downsamplings used before the data is further processed is often referred to as a *backbone*, and the features coming from such a backbone might be useful for other tasks than what the model is initially trained for. The backbone features can, for

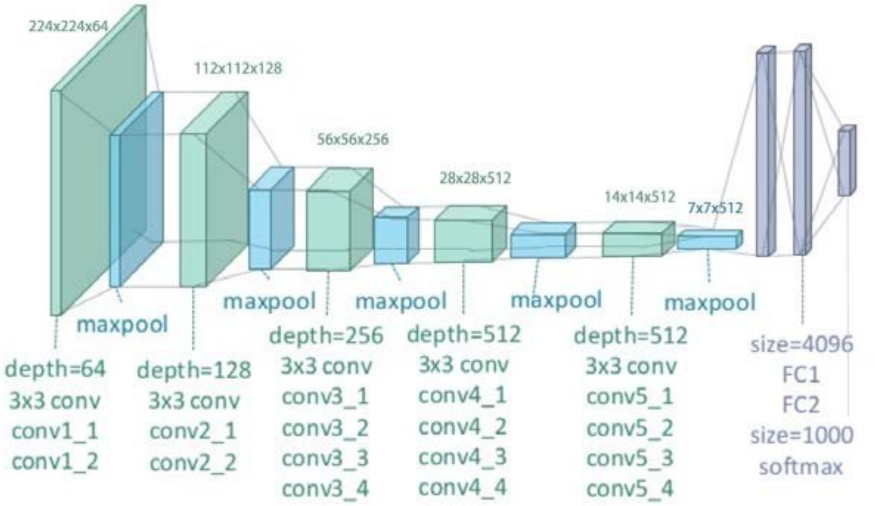


Figure 2.2: Illustration of the VGG network [11] for classification. The network takes one an image as input, processes it into smaller and smaller feature maps using convolutions and downsamplings before flattening the feature maps and doing classification using a fully-connected neural network. In the image, *conv* refers to convolutions, and *FC* refers to fully-connected network layers. Image source: [12]

example, be used as in metric for finding out how realistic images are, which we will see in section 2.13.

2.3 Object Detection using Region Proposals

Region proposal models for object detection (*e.g.* bounding box prediction) consist of two stages. The first stage tries to find possible bounding boxes, and the second stage tries to find if the possible bounding box should correspond to an object class or be classified as background.

The first model to do this was R-CNN [13], which uses the Selective Search algorithm [14] to create bounding box proposals before each of these is sent through a convolutional classification network to predict the class. R-CNN yields good results but is extremely slow as one forward pass through the entire model is needed for every bounding box. Fast R-CNN [15] makes the process faster by computing the features of the input image only once using a convolutional network. Speeding up the classification made the Selective Search algorithm the speed bottleneck of the architecture, which led to the creation of Faster R-CNN [16]. Faster R-CNN uses a separate neural network for region proposals, giving state-of-the-art performance and close to real-time inference speeds.

2.4 Instance Segmentation using Mask R-CNN

Mask R-CNN [17] is an extension of Faster R-CNN for instance segmentation. Mask R-CNN adds an additional branch parallel to the detection network of Faster R-CNN, which is used to predict the objects' masks, as seen in Figure 2.3. This design allows for accurate mask predictions while having a small overhead compared to Faster R-CNN. Pretrained Mask R-CNN models exist open-source on the internet, such as in the Detectron2 library [18].

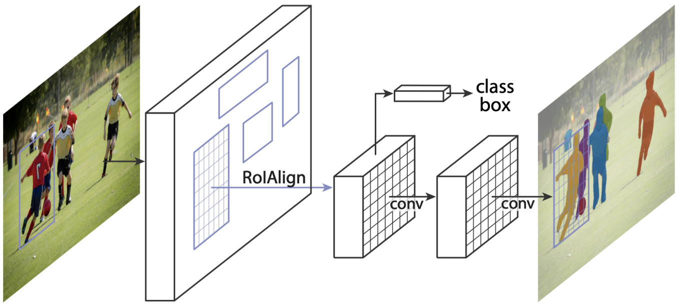


Figure 2.3: The Mask R-CNN Network builds on top of Faster R-CNN by adding a segmentation branch which is used for every region of interest found by the region proposal network. Figure source: [17]

2.5 Data Augmentation

Data augmentation is the process of creating new, realistic data from existing data so that the training set becomes more diverse. For images, this means applying image transformations preserving the labels of the original images [19]. Some typical image transformations are crop, mirror, and rotation – or all of them at once. These transformations are often applied at random when data is loaded during training, so each image becomes different each time it is loaded. Some example of image transformations can be seen in Figure 2.4.

2.6 SMPL

The Skinned Multi-Person Linear (SMPL) model [20] is a parameterized 3D human body mesh model with separate shape and pose parameters. By specifying the shape and pose parameters for the SMPL model, it is possible to generate 3D meshes of humans in a wide variety of shapes and poses, as seen in Figure 2.5. To create the SMPL body shapes, the authors align a common 3D human model

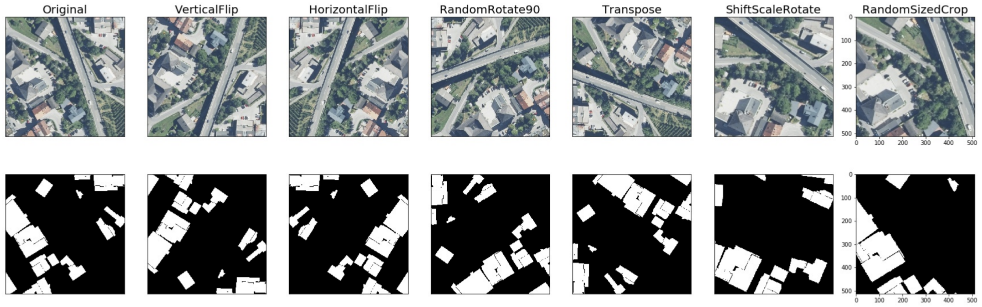


Figure 2.4: Data augmentation using image transformations. The same image transformation (text above each image) is applied to the satelite image (top row) and the ground-truth binary segmentation mask (bottom row). The goal of data augmentation is to create new images which possibly could have been present in the training set. Figure source: [19]

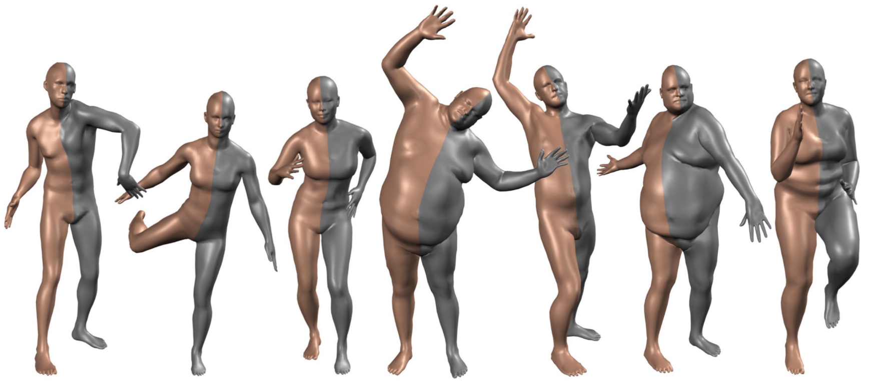


Figure 2.5: Rendered SMPL models with different shape and pose parameters. Image source: [20]

template with 6890 vertices and 23 joints to a large amount of 3D scans of real humans. These aligned 3D models are then pose-normalized and analyzed using Principal Component Analysis (PCA) to create a parameterization of the 3D human model template able to represent a large amount of human shapes with relatively few principal components.

One crucial property the SMPL model creators wanted was for the model to work with existing graphic engines. They therefore decided to do the movement of the joints by using linear blend skinning. The problem with linear blend skinning is that it tends to create unrealistic mesh deformations close to the model's joints. To counter these deformations, the authors used 3D scans of humans in a wide variety of poses to create pose-dependent shape deformations able to counter the deformations introduced by linear blend skinning.

2.7 Dense Pose Prediction

The goal of *dense pose prediction* is to predict pose information for every pixel on the human body. This is different from sparse keypoint prediction, where only single points are predicted, such as the location of all body joints. The DensePose model [21] does Dense Pose prediction by predicting pixel-to-model correspondences between pixels in the image and a SMPL model. In a sense, dense pose prediction is quite similar to instance segmentation, with the addition that all of the predicted instance pixels should contain extra information regarding where on the human body that pixel is located. The DensePose model is trained on the DensePose COCO dataset consisting of human images manually labeled with some surface points. The model outputs, for each human pixel, the corresponding body part and the UV coordinate of the body part for that particular pixel. Illustrations of these concepts are shown in Figure 2.6.

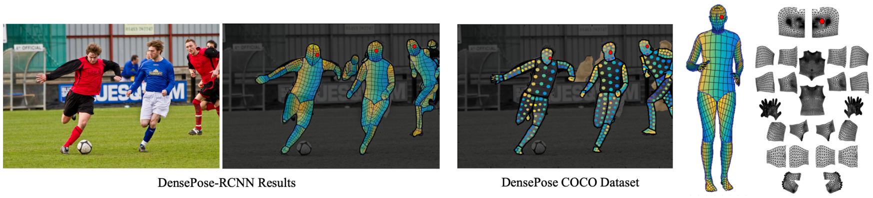


Figure 2.6: Illustration of DensePose concepts. **Left**: An image from the COCO dataset. **Middle-left**: The predicted body parts from a DensePose model where the gradient from blue to yellow corresponds to either the U or the V coordinate for each body part. **Middle-right**: Visualization of the manually annotated model points done on the image in the DensePose COCO Dataset. **Right**: The 24 body parts as they are present on the SMPL model as well as a two-dimensional UV map of all of the body parts. Image source: [21]

2.7.1 Continuous Surface Embeddings

Continuous Surface Embedding (CSE) [22] takes another approach on dense pose estimation. Instead of slicing the SMPL model into 24 parts, the model now predicts for each human pixel a vector in a learned surface embedding space in $\mathbb{R}^{16}$. This predicted vector can be used to find the closest SMPL vertex in embedding space for every pixel, thus establishing pixel-to-vertex correspondences for every human pixel in the image. The CSE model performs on-par or better than DensePose, partially because there are no “seams” between the part maps. An additional benefit of doing dense pose prediction this way is that the same scheme can be used for different deformable surface models (*e.g.* on 3D models of animals) without having to slice the models into parts. It is even possible to make a DensePose predictor work on multiple object categories by using functional maps [23] for modeling correspondences between the surface models.

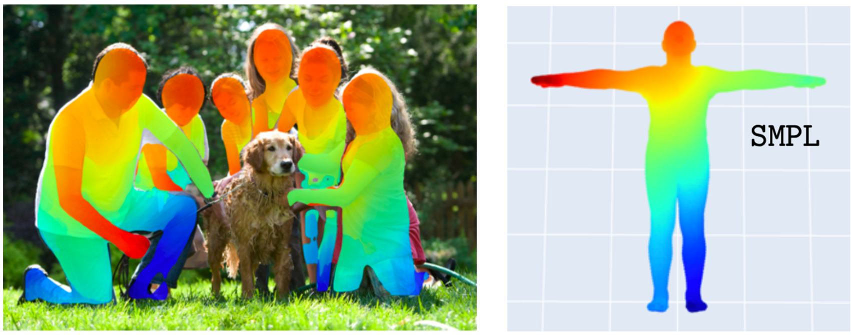


Figure 2.7: Illustration of continuos surface predictions. The model predicts a learned surface embedding value for every human pixel in the image, which in turn can be used to find pixel-to-vertex corresponences between input pixels and the vertices of an SMPL model. Each color value in the illustration corresponds to a specific SMPL vertex. Image adapted from: [22]

2.8 Editing Masks with Morphological Operations

Masks are binary images with a 0 or 1 at each spatial location and can be used to represent the location of where something exists in an image. A mask can be "multiplied" pixel-wise with another image so that everything outside of the masked area turns black. In addition, a mask is pretty easy to invert, as it can done by just applying a "logical not" operation to each pixel. To edit masks, it is possible to use operations based on mathematical morphology, such as dilation and erosion [24]. A morphological dilation is carried out by moving a *structuring element* through an image and placing a pixel everywhere the structuring element intersects with the existing mask, as illustrated in Figure 2.8. An erosion is performed much the same way, except that the whole structuring element now needs to be *inside* of the mask for a pixel to be placed in the final image. An erosion can only decrease the size of a mask, while a dilation can only increase it.

2.9 Generative Models

A generative model tries to model the distribution of the input data, in contrast to discriminative models, which try to separate the input distribution into several categories. More specifically, a discriminative model tries to calculate the probability for a label Y based on input data X, so the model's objective can be written as $P(Y|X)$. All models discussed until now, such as image classification models, Faster R-CNN, Mask R-CNN, and DensePose, can be classified as discriminative models. On the other hand, a generative model tries to model the actual distribution of the data $P(X)$, or the joint distribution $P(X|Y)$ if labels Y are provided.

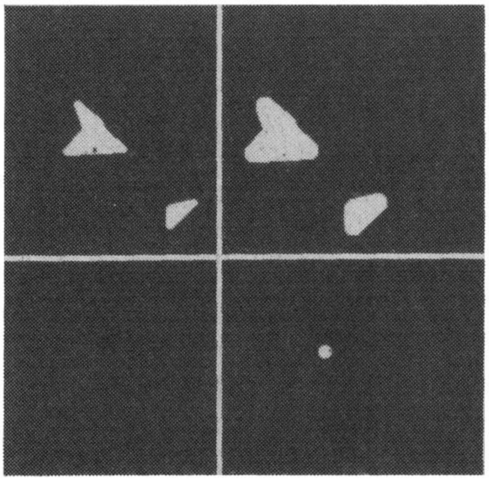


Figure 2.8: Illustration of a morphological dilation. The upper left shows the input mask, while the upper right shows the resulting mask after dilation. The dilation is done by moving the structuring element (shown in the lower right) over the input mask image and placing a pixel everywhere the two intersect. Image source: [24]

Generative models can generate new data instances or tell how likely a given data instance X is to belong to the data distribution.

2.9.1 Variational Autoencoders

An autoencoder is a model which takes data X as input and tries to reproduce the same data X as output [25]. Reproducing the output seems like a pretty easy task, but the model contains a bottleneck where the dimensionality of the throughput is severely smaller than the dimensionality of X. By introducing this bottleneck, the model must create a compressed version of the input data by simultaneously learning both a compression and a decompression function. A compressed data point (or image) is often referred to as a *latent vector*, and the space of all possible latent vectors is referred to as a *latent space*. The compression and decompression functions in autoencoders are known as encoders and decoders, respectively, and an illustration of the process can be found in Figure 2.9. It is worth noting that an autoencoder can be trained in an unsupervised manner, as no image labels are required.

Variational Autoencoders [26] is an extension of this scheme, where the latent variable is forced to follow a known distribution (*e.g*. Gaussian). By knowing the encoded distribution, it is possible to sample new data points similar to the distribution of X, as well as check whether new input images create encoded representations deviating heavily from the known latent distribution. Forcing the

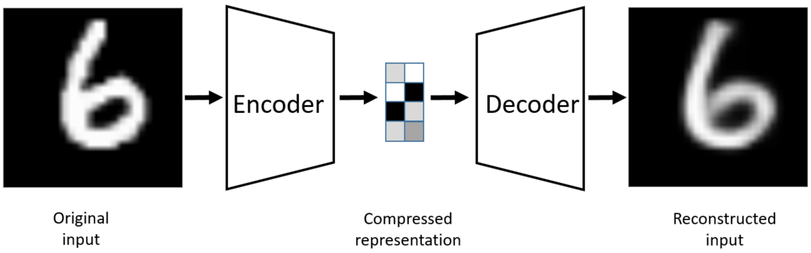


Figure 2.9: Illustration of an autoencoder for images. The image is put through an encoder and transformed into a low-demensional latent representation before being reconstructed by a decoder. The whole model it trained to do end-to-end reconstruction in an unsupervised manner. Image source: [25]

latent variable to be gaussian has also shown to make the change in output images smoother and more interpretable when moving around in the latent space. This effect can be seen in Figure 2.10, which shows samplings from a variational autoencoder latent space of two dimensions.

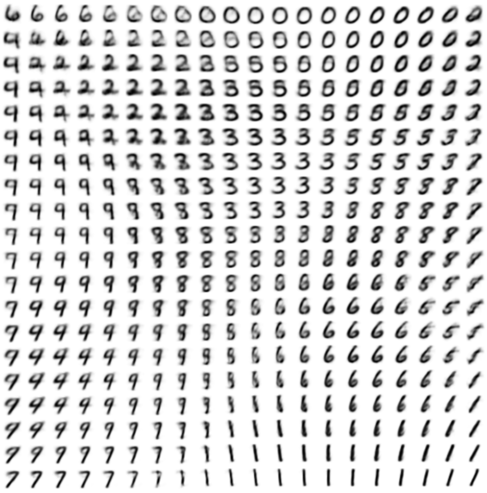


Figure 2.10: Latent space of a variational autoencoder with a 2-dimensional latent space trained on the MNIST dataset. Since the latent space is Gaussian, the points are sampled by using a grid on the inverse CDF of the Gaussian to better represent the distriubution. Image source: [26]

2.10 Generative Adversarial Networks

Generative Adversarial Networks (GAN) [27] is a framework where both a generative and a discriminative model are trained simultaneously while competing against each other. The generative model, known as the *generator*, tries to create a mapping from a multidimensional latent vector $\vec{z}$ to images looking like the ones in the training set. The discriminative model, known as the *discriminator*, tries to separate the images coming from the generator from real images in the training set. Since it is possible to compute gradients through the whole discriminator, the discriminator can be used to create targeted feedback for the generator. As training progresses, both models become increasingly good at their tasks, and after a while, the generated images will start to look much like the real ones. It is worth noting that the generator never sees any real images directly during training and is only trained by feedback from the discriminator. Calculating the loss using a discriminative model is often referred to as *adversarial loss*, or simply *GAN loss*. The optimization objective of the original GAN is given as:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z)))]. \qquad (2.3)$$

In this equation, G is a generator function converting a latent vector to an image. D is a function taking an image as input and outputting the probability of that image belonging to the real data distribution. $p_{\text{data}}(x)$ and $p_z(z)$ are the real image distribution and the latent space distribution respectively. An illustration of the GAN framework can be seen in Figure 2.11.

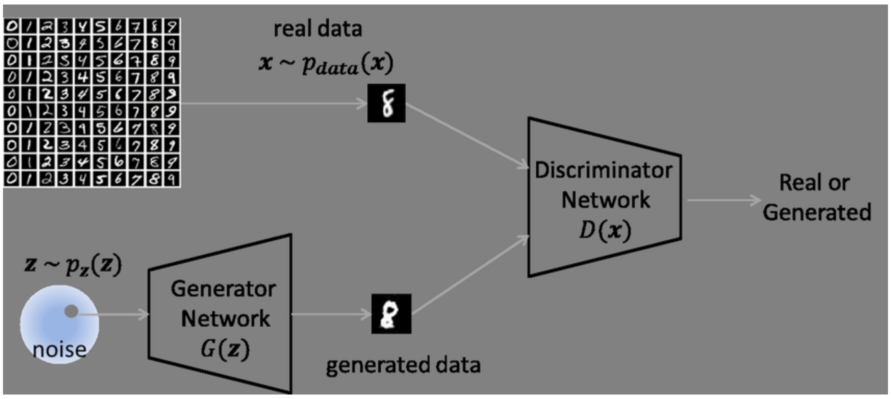


Figure 2.11: Illustration of the GAN framework. The generator uses randomly sampled noise z to generate images, while the discriminator is fed both generated and real images and tries to differentiate them from each other. The generator improves by getting targeted feedback from the discriminator. Image source: [28]

2.10.1 Wasserstein GANs

Even though the original GAN was able to create impressive images at that time, it is known for being notoriously hard to train and prone to collapsing. One of the main problems is that the discriminator can end up in a state where it is able to differentiate the real and fake samples without giving the generator good feedback. In a sense, training a good discriminator is different from training a good binary classifier, as it should not only differentiate the two classes but also provide helpful gradient values for how the generated samples can become more real.

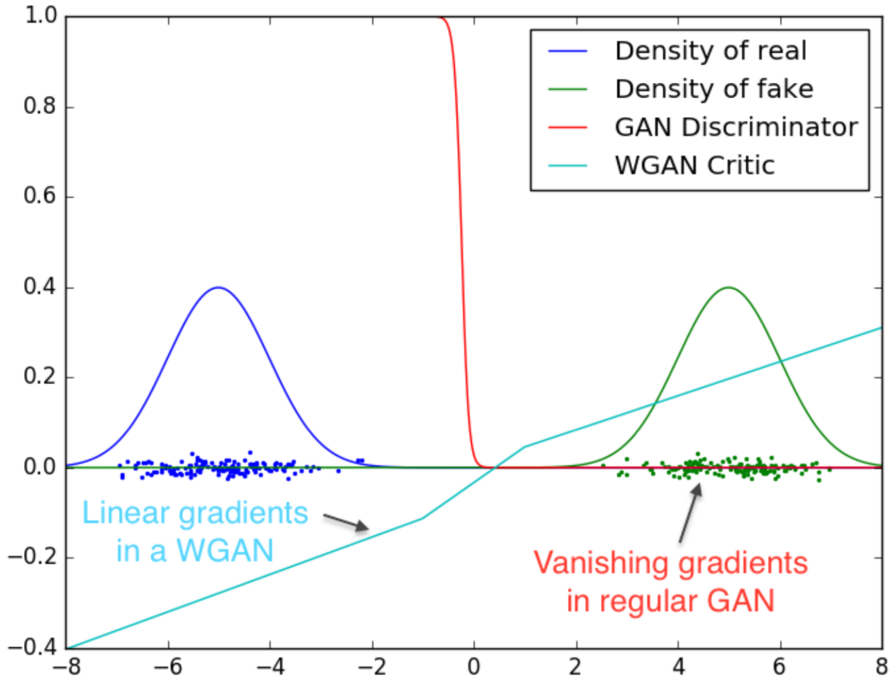


Figure 2.12: Gradients for an optimal WGAN disciminator compared to an optimal original GAN discriminator when learning to differentiate two 1-dimensional Gaussians. The original GAN discriminator ends up having a gradient close to 0 for the generated samples, which means that the generator will not be able to learn how it should change. Image source: [29]

Arjovsky et al. [29] propose the *Wasserstein GAN*, which instead of letting the discriminator output a value between 0 and 1, extends the output to all positive and negative values. It also states the discriminator outputs should be forced to be K-Lipschitz continuous, meaning that the discriminator should satisfy the criteria $|D(x_1) - D(x_2)| \leq K|x_1 - x_2|$ for all x_1 and x_2 and a positive real-valued constant K. This essentially means that there should be a speed limit on how much the discriminator output values are allowed to change, inhibiting the creation of very steep parts in the discriminator's output space. The Wasserstein GAN enforces Lipschitz continuity by using weight clipping in the discriminator, meaning that the weight values of the discriminator are clamped to $[-c, c]$ for some constant c. A comparison between the original GAN discriminator and the Wasserstein GAN discriminator trained on a 1-dimensional toy example can be seen in Figure 2.12. Note that Arjovsky references the WGAN discriminator as a *critic* since it tries to assess the “realness” of images instead of classifying them as real or fake.

2.10.2 Gradient Penalties

Arjovsky et al. state in their paper that “Weight clipping is a clearly terrible way to enforce a Lipschitz constraint”. This is partly because weight clipping pushes the weights towards the extremes of the clipping range, yielding functions that are too simple [30]. Gulrajani et al. [30] instead propose to use gradient penalties for enforcing Lipschitz-continuity of the generator. The gradient penalties work by penalizing the norm of the gradient of the discriminator with respect to its input. The new GAN objective function with gradient penalties is shown in Equation 2.4.

$$L = \underbrace{E_{\tilde{x}\sim\mathbb{P}_g}[D(\tilde{x})] - E_{x\sim\mathbb{P}_r}[D(x)]}_{\text{WGAN objective function}} + \underbrace{\lambda\mathbb{E}_{\hat{x}\sim\mathbb{P}_{\hat{x}}}[(||\nabla_{\hat{x}}D(\hat{x})||_2 - 1)^2]}_{\text{Gradient penalty}} \tag{2.4}$$

Since enforcing the gradient penalty everywhere is not possible, $\mathbb{P}_{\hat{x}}$ is approximated by uniformly sampling along straight lines between pairs of points sampled from the data distribution $\mathbb{P}_r$ and the generator distribution $\mathbb{P}_g$. This experimentally results in good performance. Mescheder et al. [31] have later proposed a similar gradient penalty, known as R_1 regularization, which works well when only performed on real images.

2.10.3 Averaging the Generator Parameters

Yaz et al. [32] found that using an average of the generator parameters yielded more realistic and clean images. The motivation behind averaging is that the generator oscillates quite a bit during training and might not manage to converge completely to an optimum. The authors try to average the generator using both a *moving average* and an *exponential moving average* and found that exponential moving average works best. The exponential moving average is defined as

$$\theta_{EMA}^{(t)} = \beta\theta_{EMA}^{(t-1)} + (1-\beta)\theta^{(t)}, \tag{2.5}$$

where $\theta^{(t)}$ are the generator parameters at timestep t, $\theta_{EMA}^{(t)}$ is the averaged generator parameters at timestep t, and β is an hyperparameter used for the averaging between 0 and 1, typically pretty close to 1. Note that the averaged generator is never used during training and just for inference.

2.10.4 Latent Spaces and Disentanglement

The z values used in the generator during training are drawn from a known distribution, most commonly a multivariate Gaussian $Z \sim \mathcal{N}(0, 1)$. The space of all possible z values is often referred to as a latent space, and the generator's job is to map points in this latent space to output images. Radford et al. [33] show that a generator during training unsupervisedly learns several higher-level representations and that interpolating between points in the latent space will create smooth transitions between the created output images. It is even possible to do vector arithmetic on the z values, as shown in Figure 2.13. The ability of a generator to learn such higher-level concepts and separate them from each other is often referred to as *disentanglement*.

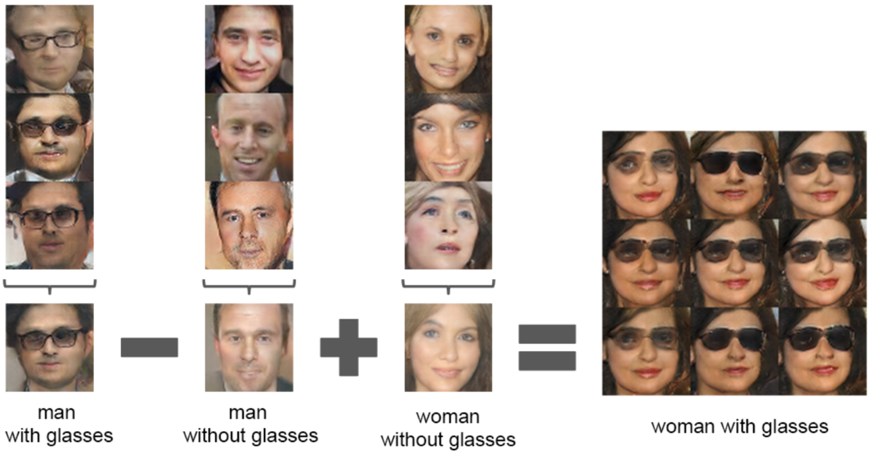


Figure 2.13: Vector arithmetic in the latent space Z. The bottom images in each column are created by an average of the z values used for the images above. Then the z values of the bottom images are combined to create the image in the center to the right. Eight other examples are generated from the center image to the right by adding uniform noise sampled with a scale of +-0.25. Image adapted from: [33]

2.10.5 Style-Based Generators

The StyleGAN model [34] created by Karras et al. makes a leap forward in terms of disentanglement by introducing several changes to the generator based on style transfer literature. The field of style transfer focuses on rendering images with different styles, such as converting a landscape photograph into an image looking like it was painted by Monet. For style transfer using deep learning, Huang et al. [35] found it beneficial to look at the *statistics* of feature maps when performing style transfer. Changing the mean and standard deviation of individual feature maps from the *content* image to the mean and standard deviation of the corresponding feature maps from the *style* image made it possible to do real-time style transfer for arbitrary styles. The component able to change the feature map statistics from one feature map to another is called Adaptive Instance Normalization (AdaIN) and is defined as

$$\text{AdaIN}(x, y) = \sigma(y)\frac{x - \mu(x)}{\sigma(x)} + \mu(y), \tag{2.6}$$

where x is the *content* feature map and y is the *style* feature map. StyleGAN uses much of the same logic, but instead of getting the means and standard deviations from another feature map y, the means and standard deviation values are created from $\vec{z}$, which is sampled from a standard Gaussian distribution like in a normal GAN. The new AdaIN operation for StyleGan becomes

$$\text{AdaIN}(x_i, y) = y_{s,i}\frac{x_i - \mu(x_i)}{\sigma(x_i)} + y_{b,i} \tag{2.7}$$

Where x_i is a feature map and y is a style consisting of a standard deviation and mean (bias) y_s and y_b respectively. By doing this AdaIN operation at every layer in the generator upsampling process, the style values y end up making more meaning, as they represent changes done at different *scales* in the generation process. To create the y values from $\vec{z}$, $\vec{z}$ is first sent through a fully-connected mapping network f to create an intermediate latent vector $\vec{w}$ which is then put through learnable affine transformations to generate the y values. In addition, noise scaled by learnable scaling factors is added to each channel individually after every convolution. The StyleGAN authors argue that adding noise makes it easier for the generator to model stochastic variation (such as the exact placement of hairs and freckles) without having to generate pseudo-random numbers from earlier activations. The whole StyleGAN architecture can be seen in Figure 2.14. Note that the input to the StyleGAN generator is a learnable constant vector that is not dependent on $\vec{z}$.

To ensure the StyleGAN model does not assume adjacent styles to be correlated, the authors employ a regularization technique they call *style mixing*. Style mixing means two sets of styles are used during the generation process for some percentage of the images during training. More specifically, two vectors $\vec{z}_1$ and $\vec{z}_2$ are sampled and used to create two vectors $\vec{w}_1$ and $\vec{w}_2$, where $\vec{w}_1$ is used up to a certain point in the generator, and $\vec{w}_2$ is used from there on and out. This way, the model is trained to be able to combine multiple style values. Some results of combining different styles from different $\vec{z}$ vectors are shown in Figure 2.15.

The authors of StyleGAN later found some problems in their original model, which led to the introduction of StyleGAN2 [36], having several improvements from the initial version. Firstly, StyleGAN2 features improved feature map modu-

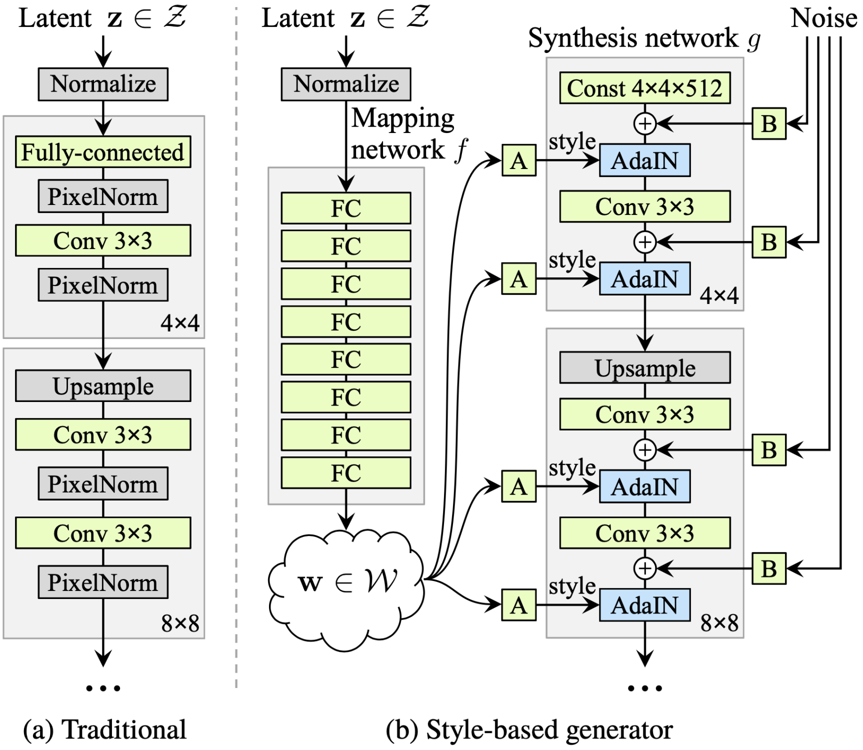


Figure 2.14: The StyleGAN generator architecture in comparison to a traditional generator architecture. For StyleGAN, the latent variable $\vec{z}$ is mapped to an intermidiate latent vector $\vec{w}$ through a fully-connected mapping network f. $\vec{w}$ is then transformed to style parameters y used in the AdaIN operations through learnable affine transformations A. Noise is added to each channel individually after every convolution with the learnable scaling factors in B. Image source: [34]

lation to remove common blob artifacts. Secondly, skip connections in the generator were added to move away from progressive growing [37], which was found to make objects “stick” to the same location when interpolating between latent vectors. Finally, a path length regularizer was added to encourage the generator to behave such that interpolating between points in the latent space creates consistent changes to the generated images.

2.11 The Truncation Trick

The *truncation trick* [38] is a technique used during GAN inference where the quality of the generated images is improved at the cost of variety. The technique is performed during the creation of the latent variable $\vec{z}$, where the values drawn above a certain threshold are resampled. This technique creates more realistic and

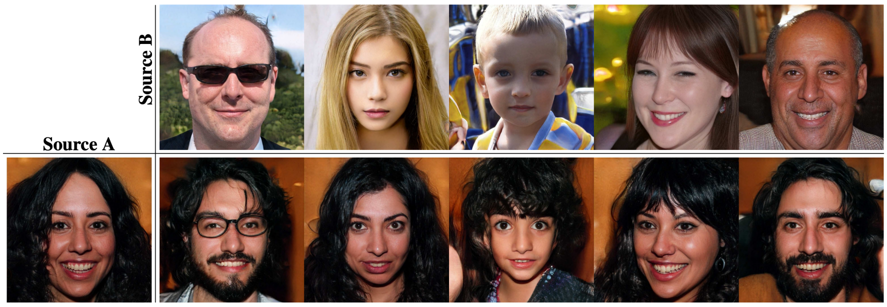


Figure 2.15: Example of StyleGAN disentanglement. The leftmost image and the top row are each generated by sampling latent codes $\vec{z}$ from a normal distribution, while the rest of the images are combinations of A and B. To create the combinations, the coarse styles (resolutions 4^2 to 8^2) from B are used while the rest of the styles are taken from A. Image adapted from: [34]

less diverse images because the generated $\vec{z}$ vectors are closer to the "center" of the data distribution and further away from outliers. The amount of truncation done during sampling is often specified by a variable t where $t = 1$ means no truncation and $t = 0$ means maximum truncation, effectively setting $\vec{z}$ to the zero vector.

2.11.1 The StyleGAN Multimodal Truncation Trick

Mokady et al. [39] propose an extended version of the truncation trick for StyleGAN by approximating local distribution hotspots for the intermediate latent $\vec{w}$ vectors. While the $\vec{z}$ vector is normally sampled using a standard Gaussian, the $\vec{w}$ vector can be of every possible distribution, possibly heavily multimodal. The "hotspots" for the intermediate latent $\vec{w}$ distribution are found by sampling a lot of $\vec{w}$ vectors for randomly drawn $\vec{z}$ vectors and computing a KMeans clustering for all $\vec{w}$. A $\vec{w}$ vector can now be moved towards the closest local clustering point instead of a global mean when truncating. This extended truncation trick is suited where $\vec{w}$ is multimodal and can be used to increase image quality with a lower drop in diversity than when using the standard truncation trick.

2.11.2 Conditional GANs

Conditional GANs are GANs using additional label information for the generation process. An early example of this is from Mirza et al. [40] which trains on the MNIST dataset [41] containing 28*28 greyscale images of handwritten digits. The MNIST label, representing which digit between 0 and 9 is present in the image, is added along with the z value as input to the generator and is also added to the discriminator along with an image.

2.12 Image-to-Image Translation

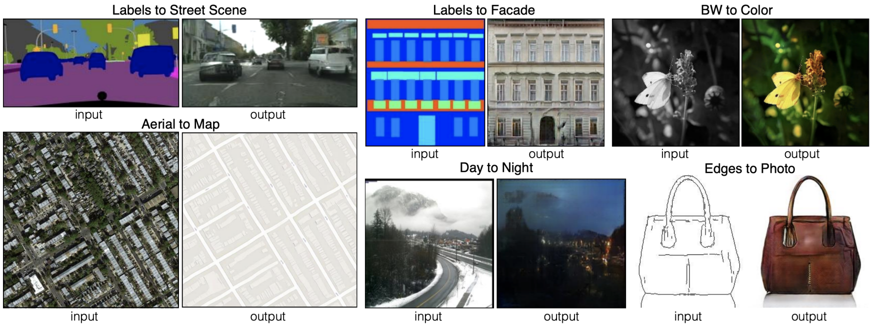


Figure 2.16: Examples of various image-to-image translation tasks, such as labels to image, aerial image to map, recolorization, day to night and edges to photo. Image source: [42]

Image-to-image translation is the task of transforming images of one domain into images of another domain. Some common use-cases for image-to-image translation, in the paired case, are shown in Figure 2.16. These tasks are recognized by the input and output images having structural similarities. However, the images might differ in the number of channels and the range of values in each channel. In this sense, the task of semantic segmentation (discussed in section 2.1.1) can be seen as an image-to-image translation task. Another paired image-to-image translation task is *image inpainting* [43], which is the task of filling in missing regions of an image. Work exists on unpaired image-to-image translation [44] as well, such as transforming images of horses to images of zebras, but this is much more difficult as no ground truth image translations are available.

One of the first models used for general paired image-to-image translation is Pix2Pix [42], which can be seen as a conditional GAN where an input image is used as conditioning for the generator. The discriminator's job is to take in a pair of images and check if they constitute a real pair or if one of them is generated. The reason pairs of images must be supplied to the discriminator is that this ensures that the generator learns a paired mapping and is not free to generate *any* real image. An overview of the Pix2Pix architecture can be found in Figure 2.17.

The reason Pix2Pix is revolutionary is that it can handle a lot of paired image-to-image translation tasks quite well. While some other work focuses on only one of the tasks, such as recoloring [45], Pix2Pix is very versatile and does not require a hand-crafted loss function for every new task. This is because the GAN discriminator can be seen as a "learnable loss function" able to adapt to the task at hand.

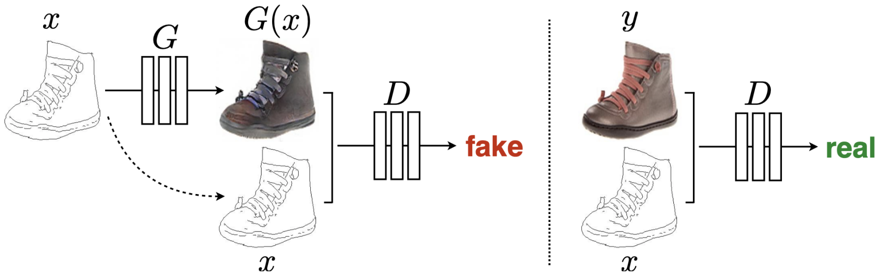


Figure 2.17: Pix2Pix architecture for the edges to photo case. The conditioning (input image) is always passed to the discriminator, so that the generator is forced to learn a paired mapping. Image source: [42]

2.12.1 Unet Encoder-Decoder Architectures

The generator in Pix2Pix employs a Unet architecture [46], which was created initially to do biomedical image segmentation. It is called Unet as the architecture resembles a U when illustrated, as seen in Figure 2.18. The Unet works similarly to the encoder and decoder of an autoencoder, with the addition of skip-connections between feature maps of the same spatial resolution. These skip connections make the paths from input to output shorter, in addition to make it easier for the network to handle the spatiality of the input content. When convolutions and downsamplings are done to an image, semantic (or meaningful) information is created, while the spatial information in the input is lost. The skip connections make sure the decoder part of the model (right side of the U) can look at the information coming from below when deciding *what* content it should contain and look at the information coming from the "opposite" side of the U regarding *where* the content should be placed. In the case of segmentation and image translation tasks in general, keeping the spatial information from the input is essential to create precise outputs.

2.12.2 Handling Multi-Modality in Image Translation

In many image translation tasks, there exists an ambiguity in how the final image should become. When doing recolorization, it might be ambiguous what color a bird should have, and when doing image inpainting of a face where one eye is missing, it might be ambiguous if the eye should be open or closed. This ambiguity is somewhat difficult to model when training with paired images as there is only one ground truth to each input image. It is possible to add noise to the input of Pix2Pix naively, but Pix2Pix will have no incentive to use the noise and will end up mostly ignoring it. Zhu et. al. [47] propose the BicycleGAN model, which makes the sure the noise is being used by adding an additional mapping E from an image

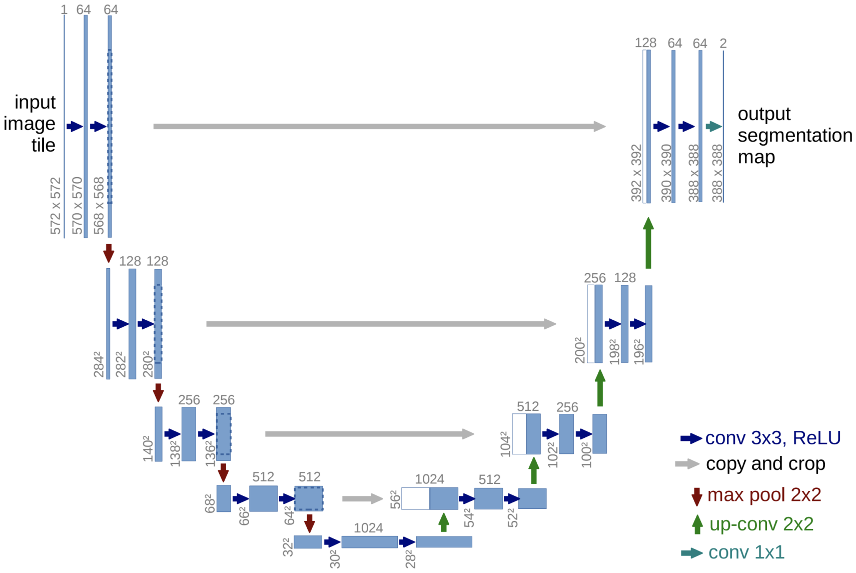


Figure 2.18: Example of a Unet architecture. Each blue box corresponds to a feature map with multiple channels, and the arrows represent transformations from one feature map to another. The white parts of the feature maps to the right represent copies from the left part of the architecture. Image source: [46]

x to the latent vector $\vec{z}$ and enforcing the cycles $G(E(x)) = x$ and $E(G(\vec{z})) = \vec{z}$. BicycleGAN additionally uses Kullback-Leibler divergence loss to force the $\vec{z}$ predicted by E to follow a normal distribution. Some results from BicycleGan can be seen in Figure 2.19.

2.13 Evaluating Generative Models

Quantitative evaluation of generative model performance has traditionally been challenging, as it is not easy to create an automated process able to judge the quality of the generated images in the general case. Most image quality metrics in use today take advantage of a pre-trained computer vision model to assess image quality. The metrics are justified in that they, in many cases, seem to coincide with human image quality judgment.

2.13.1 Inception Score

The Inception Score [48] is one of the early metrics to judge the image quality of generated GAN images. The metric uses class probabilities computed with an Inception v3 module [49] trained on the ImageNet dataset [6] when deciding on

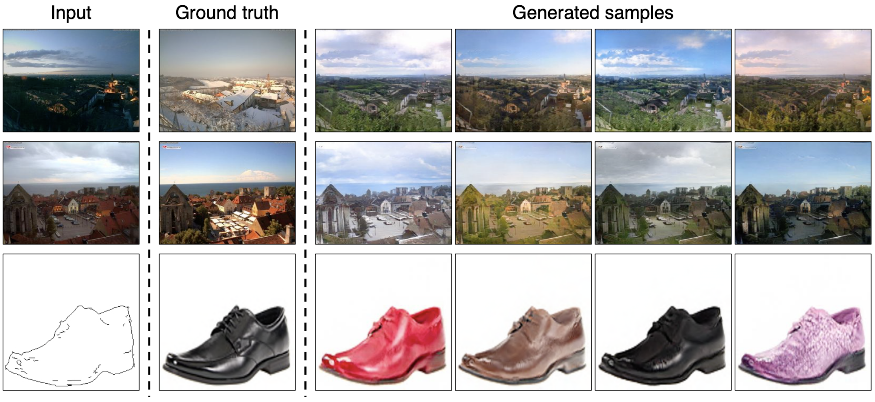


Figure 2.19: Example images from BicycleGAN in both night-to-day translation and edges-to-shoes translation. The left column shows the input image, the second column shows ground truth image pair and the rest of the images are randomly generated from the input image in their row. Image source: [47]

the quality. To get a good Inception Score, individual images should be classified as a single (or few) classes, while the compound distribution should be flat (for high variance).

To compare the distributions, the metric uses Kullback Leibler Divergence. A problem with the Inception Score is that it is most suitable for measuring the quality of images belonging to the 1000 image classes in the ImageNet dataset. The training images used during training are not considered when computing the metric at all, so for arbitrary domains, it might be better to use other metrics.

2.13.2 Fréchet Inception Distance

The Fréchet Inception Distance (FID) Score [50] is more general than the Inception Score as it uses statistics from images in the training set to compute the final metric for the generated images. The FID Score works by collecting image features (instead of class probabilities) from an Inception V3 network and calculating the mean and covariance for each feature. This process is done for both real and generated images, and the two resulting distributions are compared using the Fréchet Distance. Here, lower scores are better, as it means the model produces images close to the training image distribution in feature space.

In the FID paper [50], they partially test the metric by checking how well it reacts to various image distortions being applied to an image. Increasing the applied amount of image distortions should ideally result in increasing FID scores. An example of how the FID scores react to distortions are shown in Figure 2.20.

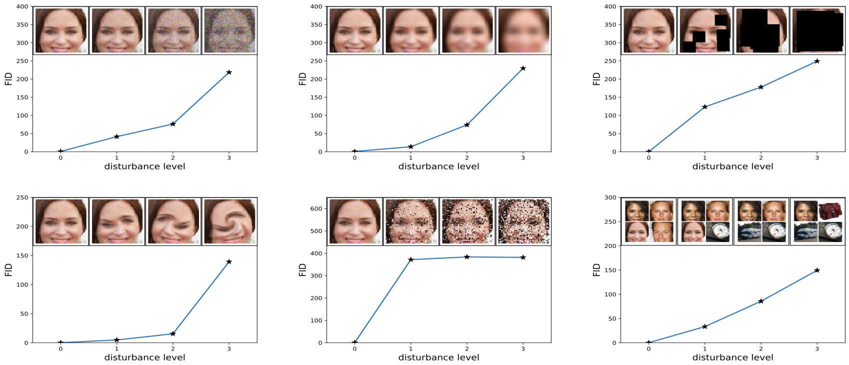


Figure 2.20: Example of how increasing amounts of image distortions results in almost monotonically increasing FID scores. The first 5 images show how the FID score reacts to image distortions while the image in the bottom right shows how FID score reacts by contaminating CelebA images [51] with ImageNet images [6]. Image source: [50]

FID CLIP

Recently, Kynkaanniemi et al. [52] have found that FID is not ideal as a general perceptual metric to compare generative models. The main problem is that the Inception model used for FID creates features that are highly sensitive to the presence of ImageNet objects. For many domains, such as face generation, relying on ImageNet features does not always correlate *that* well with improved image quality. They demonstrate this effect partially by creating an “attack” consisting of hand-picking generated images containing certain ImageNet features and showing that these images have significantly lower FID without improved image quality. In their paper, they propose to instead use features from a model trained on a more general task than ImageNet classification and decide on using a Contrastive Language-Image Pretraining (CLIP) [53] model. CLIP is trained on predicting which caption goes to which image on a set of 400 million (image, text) pairs and therefore has much more general features than the original Inception model used for FID. The new FID metric is named FID_{CLIP} by kynkaanniemi et al.

2.14 Quantifying Image Similarity

Assessing image *similarity* is important both in generative modeling and other fields. If we, for example, want to measure how good an autoencoder is at reproducing the output, we need some way of comparing the input and output images. To do this comparison using pixel-wise differences is, in general, not a good idea, as looking at each pixel individually does not take image structures into account and does not match well with human similarity judgments [54]. Only looking at the pixel-wise differences will, for example, rate a blurred version of the image as similar, while a version with some added noise might look very dissimilar. The Learned Perceptual Image Patch Similarity (LPIPS) metric [54] instead measures similarity using features from a convolutional network. The creators of LPIPS show that measuring the similarity this way corresponds better with human judgments for a wide array of image distortions. How the LPIPS metric is computed is shown in Figure 2.21.

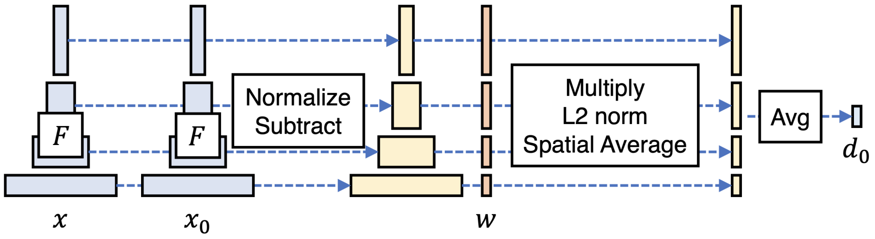


Figure 2.21: Computation of the LPIPS metrics. The two images are sent through the same pre-trained network before their features are normalized, subtracted, multiplied with weights w, sent through an L2-norm and averaged spatially. Image adapted from: [54]

The authors provide several ways of calibrating the model weights to match human quality judgments better, namely keeping the weights of the pre-trained network fixed and only changing w, do the same without fixing the pre-trained weights and training it all from scratch. When w is set to 1 everywhere, the comparison of the features is the same as measuring the cosine distance between them.

Chapter 3

Related Work

This chapter will describe how the problem of consistent full-body anonymization differs from existing solutions for anonymization and human synthesis. We will compare our task to similar tasks from recent literature and highlight the critical challenges in our problem, which are not present for other architectures and datasets. At the end of the chapter, we will create a small framework for describing different types of variation present in human datasets before comparing some of the existing datasets using this framework.

3.1 From Face Anonymization to Full-Body Anonymization

Several systems exist for face anonymization. DeepPrivacy [55] uses GANs conditioned on face keypoints to generate new faces for a given image context. Conditional Identity Anonymization Generative Adversarial Networks (CIAGAN) [56] takes this process one step further regarding controllability by providing an input identity into the anonymization process. The CIAGAN model is trained to make the final identity looks closer to that of the given input identity, improving controllability. It is worth noting that CIAGAN cannot regenerate the desired identity completely, as the rest of the person's head will look the same. Examples of generated results of CIAGAN are shown in Figure 3.1.

When moving from face anonymization to general full-body anonymization, several new problems arise:

- **No local context**: Face anonymization models observe parts of the original identity (*e.g.* contour of the face). In contrast, full-body anonymization ob-

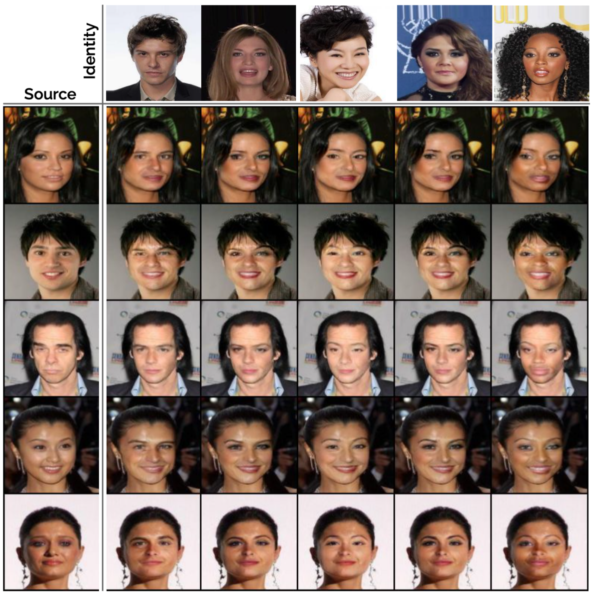


Figure 3.1: Examples of anonymizations from CIAGAN. The left column shows the source image from anonymization and the top row shows the desired identity. The final images become a mix of the two. Image source: [56]

serves no part of the original identity, meaning the context must be solely determined by the "background" pixels.

- **Harder edges**: The edges of the regions to be anonymized will always be connected to the background, making generating realistic edges harder.
- **More variation**: The shapes of the regions to anonymize will have a lot more variation, especially when it comes to hair and body pose
- **Environment interaction**: Images of people, in general, contain much more interaction with the environment and objects, as well as interaction with other individuals and occlusions.

These problems combined make full-body anonymization quite a challenge compared to the anonymization of faces. Especially if we want to anonymize images of all possible kinds, commonly referred to as *in-the-wild* images.

3.2 Full-Body Anonymization

The authors of CIAGAN demonstrated their model to work for full-body anonymization. However, their method is trained for low-resolution images without high-frequency details. Surface Guided GANs [57] increases the resolution and introduces pose information gathered from a CSE model (subsection 2.7.1) when anonymizing. The pose information guides the model regarding where the different parts of the human body should be placed, as seen in Figure 3.2. Surface-Guided GANs aim to create realistic and diverse anonymization fitting the given image context. However, the model relies a lot on background information when generating people, which can lead to bad performance in terms of temporal consistency between image frames.

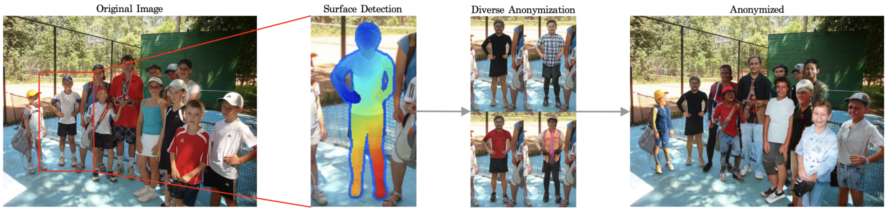


Figure 3.2: Illustration of the two-stage anonymization process of Surface-Guided GAN. Each person is detected individually by using a CSE model (subsection 2.7.1), before the person pixels are removed and a conditional GAN creates a new person in its place by utilizing the CSE information. Image source: [57]

3.3 Full-Body *Synthesis*

There is a limited amount of work on full-body anonymization, but more work exists on full-body *synthesis*, where the goal is to generate human images from scratch. The task of full-body synthesis differs from anonymization in that the model will not have a given image context (*e.g.* background) that the generated person needs to match. Pose information is widely adapted for full-body synthesis to guide the model when generating new images. For example, “Pose Guided Person Image Generation” [58], use 18 human keypoints to guide the image generation process. Similarly, “Dense Pose Transfer” [59], uses denser pose information from a DensePose [21] model for image generation, partly by warping an extracted DensePose texture to a new pose.

A common goal for full-body synthesis is that the input parameters used for image generation are *disentangled*. In this context, “disentangling” refers to a system’s ability to change certain aspects of the generated person without changing other aspects in the process. For example, a disentangled full-body synthesis system should be able to recreate the same person in a different pose without changing the person’s clothes, face, or hairstyle. Current work in this field includes Disentangled Person Image Generation [60], which continues the work on Pose Guided Image Generation by trying to explicitly disentangle foreground, background, and pose. Similarly, StylePoseGAN [61] uses dense pose representations for disentangled human synthesis. StylePoseGAN, by utilizing DensePose information as conditioning, can explicitly disentangle pose and appearance quite well for high-fidelity output images (see Figure 3.3). StylePoseGan is trained on paired data from the DeepFashion dataset (further described in section 3.4) and uses a combination of GAN loss and paired image reconstruction loss during training.

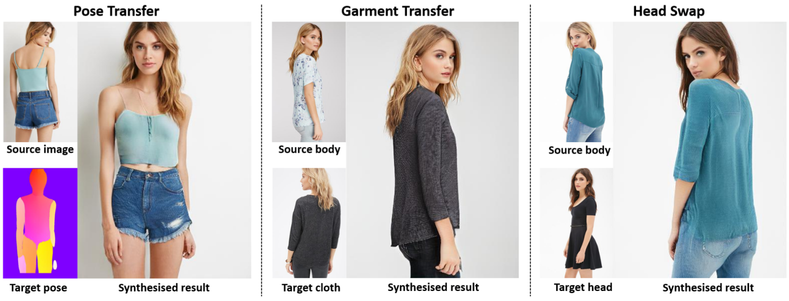


Figure 3.3: Example results from the StylePoseGAN paper. The StylePoseGan model is trained on paired images from the DeepFashion dataset and can explicitly disentangle pose and appearance during the generation process, opening for tasks such as pose transfer, garment transfer and controllable human synthesis. Image source: [61]

Note that the aforementioned models' goal is not anonymization but rather other tasks such as pose transfer, image interpolation, virtual try-on, and controllable human generation. As these models are trained for creating images from scratch and not for replacing existing image content by taking the image context into account, they are not directly applicable for the task of full-body anonymization. In addition, the datasets used by these models do not contain enough variation for anonymization of in-the-wild images, as we will see in the following sections.

3.4 Datasets used for Human Synthesis

For faces, there are mainly two datasets which are used for synthesis, the CelebFaces Attributes Dataset (CelebA) [51] and Flickr-Faces-HQ (FFHQ) [34]. CelebA contains images of celebrities while FFHQ contains images scraped from the photography website Flickr. Many papers also uses an edited and higher version of the CelebA dataset, which is known as CelebA-HQ [37]. Some example images from the CelebA and FFHQ datasets are shown in Figure 3.4.

For full-body human synthesis, Market-1501 [9] and DeepFashion [62] are widely adapted. Market-1501 consists of images taken in front of a supermarket at Tsinghua University in Beijing, boasting 32 668 annotated bounding boxes of 1,501 person identities. The subjects are captured from up to 6 different cameras so that they are viewed from different angles. The DeepFashion dataset contains multiple benchmark datasets for fashion-related computer vision tasks, with the dataset most used for human synthesis being the *In-shop Clothes Retrieval* dataset, containing 50 000 studio images of people in various kinds of clothing. When

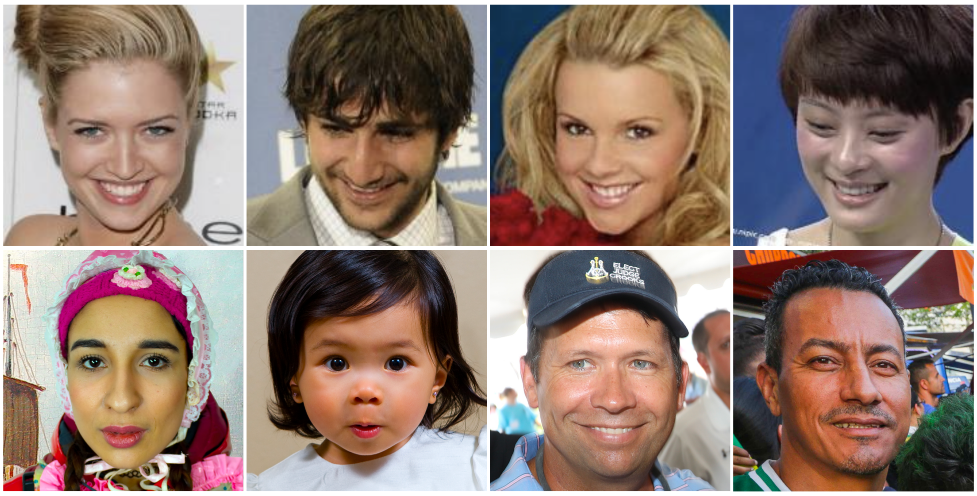


Figure 3.4: Example images from the CelebA dataset (top row) and FFHQ dataset (bottom row).

mentioning DeepFashion later in this thesis, we will refer to the In-shop Clothes Retrieval benchmark dataset and not the rest of DeepFashion. Figure 3.5 shows examples from these datasets.

Both the Market-1501 and DeepFashion datasets contain identity mappings of the subjects in the images, making it possible to create pairs of images with the same person. Having paired images is an advantage for some human synthesis tasks such as pose transfer, where the goal is to recreate a given person in a new pose. By using paired data for pose transfer, it is possible to compare the generated image to a ground truth image, for example by using similarity metrics such as perceptual loss [63]. Having paired data in general open for more disentangled representations of identity and pose. However, DeepFashion and Market-1501 lack diversity. Both datasets are taken from a small domain of all the possible images of people and are therefore not suited for full-body anonymization of in-the-wild images. The variation is not that large, there are no examples of people interacting with objects present in the images, as well as little diversity in background and few examples of occluded people. Further analysis of the different types of variation present in the human datasets will come in section 3.6.

3.5 The COCO-Body Dataset for Full-Body Anonymization

The COCO-Body dataset was introduced along with Surface-Guided GANs [57]. COCO-Body is generated by using a CSE model to create pixel-to-model correspondences on person images from the COCO dataset. The pixels which are to be anonymized for each person are found by applying a morphological dilation to the

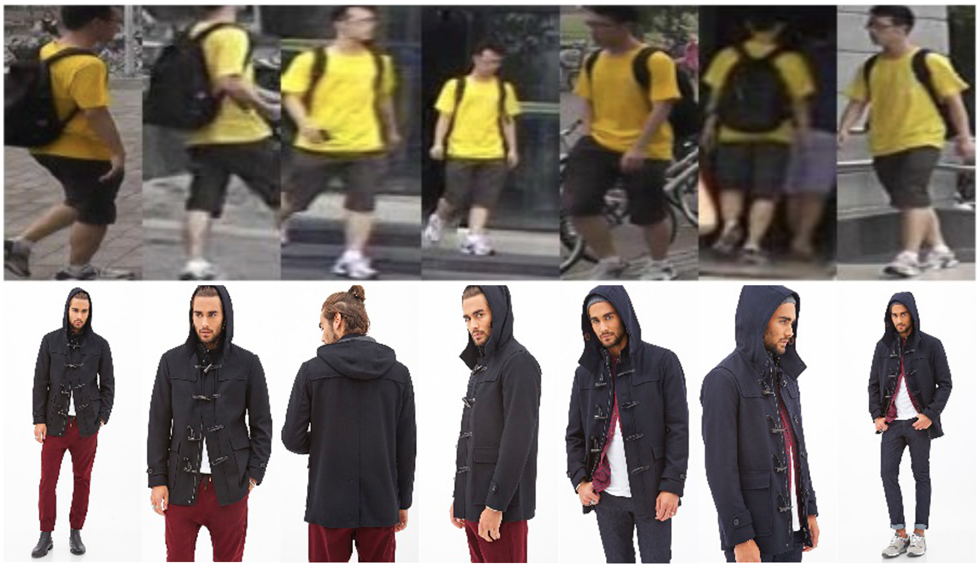


Figure 3.5: Example images from the Market-1501 dataset (top row) and Deep-Fashion dataset (bottom row).

mask of all detected CSE pixels. The reason for applying the dilation is to try to ensure the mask covers the whole body. As the images in the dataset are gathered from COCO images, the COCO-Body dataset is highly diverse but does not have paired data. Some examples of images in the COCO-Body dataset can be found in Figure 3.6. Later in this thesis will use a dataset somewhat similar to COCO-Body named Flickr Diverse Humans (FDH), which is more thoroughly described in section 4.3.

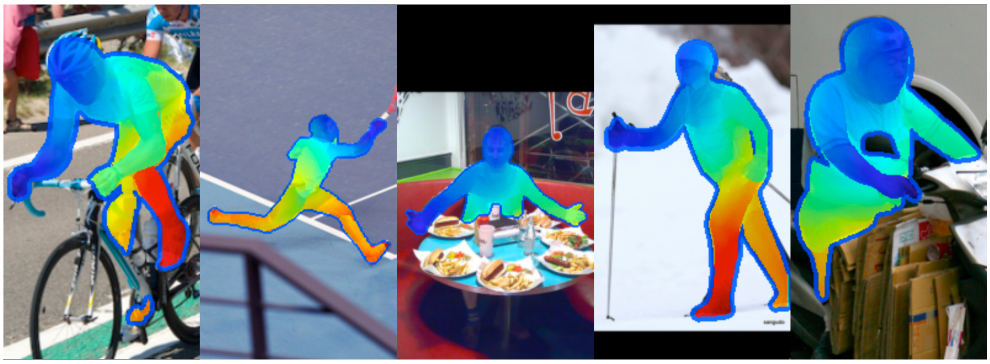


Figure 3.6: Example images from the COCO-Body dataset. The colorful overlay represents pixel-to-vertex correspondences from CSE (subsection 2.7.1), and the outer blue edge is the dilated CSE region which represents the pixels which are to be changed in the anonymization process. Image adapted from: [57]

3.6 Factors of Variation in Human Datasets

The datasets mentioned throughout this chapter all have different degrees of variation, but they are varied in different ways. To be able to compare these datasets in terms of their variation, we have created a simple framework separating the types of variation into the broad categories of *pose*, *appearance*, and *context*. A dataset's variation in each of these "factors of variation" will significantly impact how a model trained on such a dataset will be able to do full-body anonymization for in-the-wild images. The definitions we have given for variation in pose, appearance, and context will be outlined below.

Variation in Pose

Variation in pose describes how the human body is portrayed in the images. This includes both the variation in the body poses of the people involved, as well as variation in viewpoint and body occlusions. Note that this definition of pose is different than in the SMPL model domain (section 2.6), where pose is solely restricted to the rotation of body joints.

Variation in Appearance

Variation in appearance refers to the variance in how people in the images look. This includes variation in gender, age, ethnicity, clothing, hair color, hairstyle, hats, and so on.

Variation in Context

Variation in context refers to variation in the remaining possible factors that influence how the person in the final image will appear. This includes lighting (both the direction and if the light is hard or soft), exposure, white balance, blur, contrast, colored light, Instagram filters, and probably a lot more. It is reasonable to assume that the context will be similar between subject and background, but there might be a slight mismatch in some cases.

3.6.1 Human Image Dataset Comparison in Terms of Pose, Appearance, and Context Variation

Table 3.1 compares the datasets mentioned throughout this chapter for faces and full bodies. For each of the factors of variation from section 3.6, we give each dataset a score of *low*, *medium* or *high* by qualitatively inspecting each dataset. Note that these ratings are highly subjective and only presented to give the reader a quick overview of the datasets. When we refer to “paired data” in Table 3.1, we do not mean that the dataset is organized into pairs but that multiple images are present for each identity, such that the construction of pairs is possible.

Table 3.1: A comparison of the widely adapted datasets for human synthesis and anonymization. The definitions of pose, appearance, and context are described in section 3.6. The rightmost column tells if the dataset has identity mappings for the people present, so that it is possible to get paired images of the same person.

	variation in pose	variation in appearance	variation in context	contains paired data
CELEB-A	low	medium	medium	yes
FFHQ	low	high	medium+	no
Deepfashion	medium	medium	low	yes
Market-1501	medium	medium	medium	yes
COCO-Body	high	high	high	no

The datasets only containing faces, namely FFHQ and CELEB-A, are given low variance in pose as they are all seen from the front of the person with no occlusion. The FFHQ dataset is rated higher for appearance, as it has a larger diversity in terms of age, ethnicity, clothing, and so on for the people involved. In addition, FFHQ is rated a bit higher for context as the amount of settings, types of lighting, and cameras used seems to differ more.

Regarding the datasets containing entire bodies, COCO-Body is rated higher in all categories, as it includes images with high diversity in pose, appearance, and context. This is definitely not the case for DeepFashion and Market-1501.

3.7 Conclusion on Related Work

To sum up, while quite some work exists for face anonymization and controllable face anonymization, the field of full-body anonymization is still in its infancy, with no work present for consistent full-body anonymization as far as we know. Current systems for full-body *synthesis* are able to disentangle pose and appearance quite well, but these systems are not directly applicable to in-the-wild anonymization. This is due to the fact that full-body anonymization systems are not designed for inpainting regions with a given background or handling image domains with large variations in context and pose. In addition, image synthesis models rely heavily on paired data to make them controllable.

Chapter 4

Method

This chapter will cover the dataset, model architectures, loss functions, and metrics used for the experiments in chapter 5. We will start by discussing desirable properties of a consistent anonymization system before introducing the FDH dataset used to train all our models. Then, we will describe our baseline model based on StyleGAN before elaborating on the changes done to the model architecture and loss function to make the anonymization process more consistent. At the end of the chapter, we will describe the metrics which will be used to assess various aspects of the final system.

4.1 Desirable Properties of a Consistent Full-Body Anonymization System

For a consistent full-body anonymization system, the synthesis method should ideally be able to insert any person in any given place for any given pose. In order to achieve this, the model should be capable of disentangling appearance from context and pose, such as these concepts are described in section 3.6. The generated person should have the same pose as the original person, and the context of the generated pixels should match that of the original image. Ideally, only the *appearance* of the generated person should change.

4.2 Choosing the Dataset

When choosing which dataset to use for anonymization, there is a heavy trade-off between having diverse images or paired images, as described in subsection 3.6.1.

If the model is to work for in-the-wild images, large variation is needed, but having paired data might make it easier to create disentangled representations. In this thesis, we decided to investigate how to make a model trained on unpaired images more consistent rather than settling for a system only able to work in a limited number of cases for current paired datasets. Based on this decision, we decided to use a dataset with as much diversity as possible, namely the FDH dataset, which is presented in the following section.

4.3 The FDH Dataset

The Flickr Diverse Humans (FDH) dataset is a new large dataset for full-body synthesis of in-the-wild-images. FDH has much the same format as the COCO-Body dataset described in section 3.5 but features increased diversity, a larger amount of images, and better masks of the areas to be anonymized. The FDH dataset was made mainly by my supervisor Håkon Hukkelås for other purposes during the thesis period, with me contributing with some analysis and filtering, which can be seen in Appendix A, as well as some discussion. The dataset is not mentioned in current literature, so the following sections will provide a detailed description of the content of FDH, the FDH dataset creation pipeline, and the reasoning behind creating the dataset this way. Some examples of FDH images can be seen in Figure 4.1.

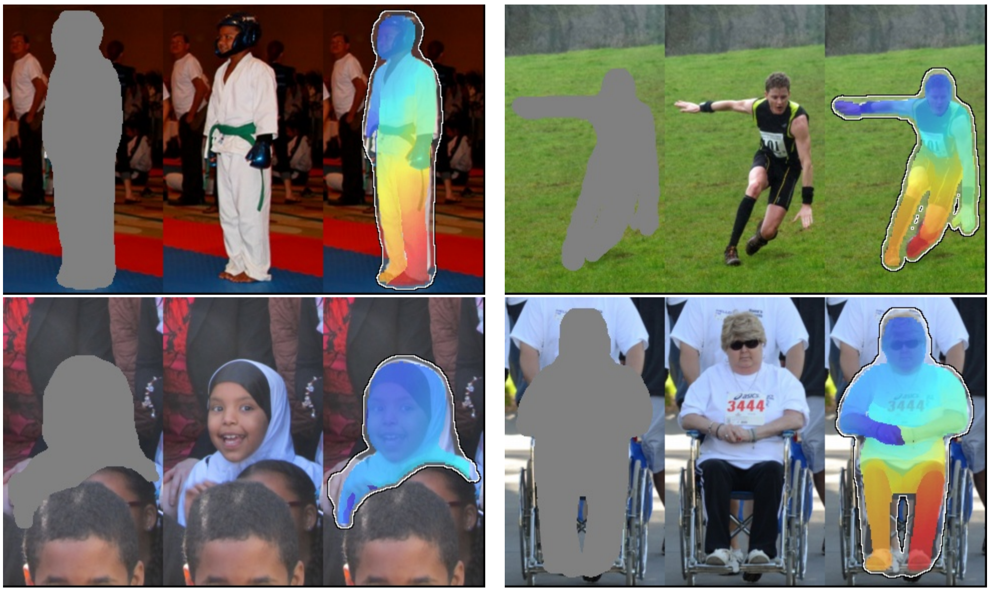


Figure 4.1: Some example images from the FDH dataset with illustrated person mask and pose information. From left in each group of images: Area to be anonymized, original image, CSE prediction with surrounding person mask.

4.3.1 Base dataset

The FDH dataset is essentially a processed and filtered version of the Yahoo Flickr Creative Commons 100 Million Dataset (YFCC100M) [64], which is a much larger and more diverse image dataset than the COCO dataset used for COCO-Body. The YFCC100M dataset, as the name suggests, consists of 100 million Flickr images with Creative Commons licenses for research purposes. Flickr is a popular photography website with a lot of different users from all over the world, so the YFCC100M dataset contains images with a large variety of backgrounds, perspectives, and lighting conditions. The people featured in the dataset have a large diversity in pose, clothing, ethnicity, body shapes, occlusions, and so on. Note that basing the FDH dataset on YFCC100M means that FDH will inherit all the biases which might be present in YFCC100M.

4.3.2 The FDH Dataset Creation Pipeline

The creation of the FDH dataset is done automatically by using existing prediction models in the following processing pipeline: First, an YFCC100M image is sent through both a Mask RCNN model and a CSE predictor. Second, the pipeline checks for overlapping predictions for the two models. For cases where the predictions have sufficient overlap, an image of size 288 × 160 is cropped around the human, and both the mask and CSE prediction are stored. Creating the dataset this way ensures that all final images have one - and only one - person as a subject, even though other people close to the subject can exist in the images. The process is illustrated in Figure 4.2.

The reason both the Mask RCNN model and the CSE prediction model are needed is that these models, even though they both detect humans, are trained for different purposes. The CSE model is trained to predict model-to-vertex correspondences and thus disregards some human parts we want to anonymize, such as clothes and hair. The Mask R-CNN model, on the other hand, is better at predicting masks covering irregular clothes and hair but does not provide semantic information as CSE does. Combining predictions from both models makes it possible to get both good pose and mask information. An example where the mask from Mask R-CNN makes a better mask than the pixels from the CSE prediction can be seen in Figure 4.3. We have decided to call the mask from Mask R-CNN a *person mask* in the context of the FDH dataset.

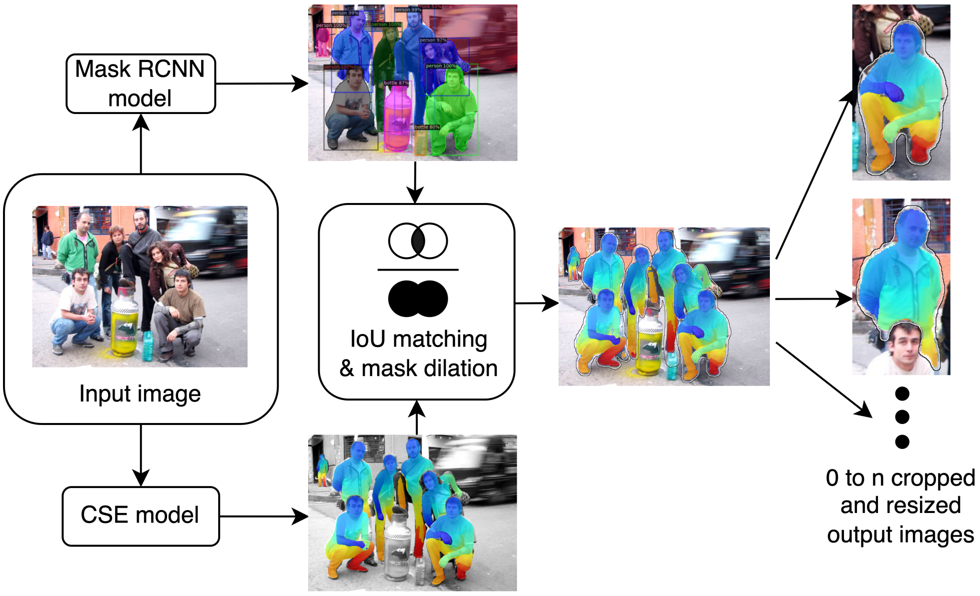

Mask RCNN model
Input image
IoU matching & mask dilation
CSE model
0 to n cropped and resized output images

Figure 4.2: The FDH dataset creation pipeline. An input image is run through both a CSE model and a Mask RCNN model before their outputs are combined based on an IoU threshold. From this combined output, one image is cropped and saved for each detected person.

4.3.3 Final Dataset Statistics

The final version of the FDH dataset contains 1 970 803 training images and 30 000 validation images. For each of the images, both the image data, a person mask, a CSE mask, and the pixel-to-model correspondences from CSE are given. It is worth noting that the raw CSE embedding values are not stored but rather the vertex-correspondences for each pixel, meaning that there can be some clusters of pixels in the CSE prediction having the same value.

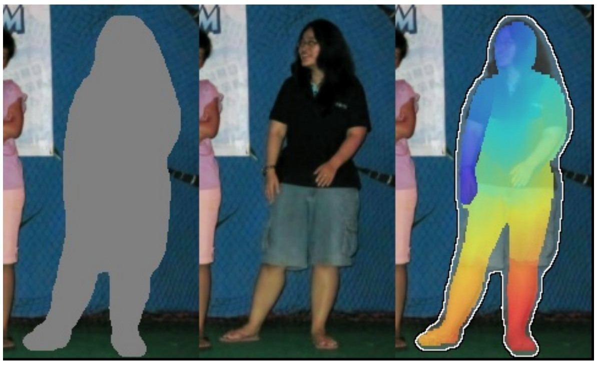


Figure 4.3: An example image from the FDH dataset illustrating the importance of using both a CSE predictor and a Mask RCNN model. With a dilated CSE mask as used in the COCO-Body dataset, the baggy clothes and long hair would not have been sufficiently masked out. From left: Area to be anonymized, original image and CSE prediction with surrounding person mask

4.4 Baseline Architecture Based on StyleGAN

The baseline model, to which we will be comparing our experiments, is a StyleGAN architecture (subsection 2.10.5) adapted for image-to-image translation. The baseline is illustrated in Figure 4.4 and differs from a standard StyleGAN in that it has an added encoder in front of the generator. This encoder enables the generator to consider the background pixels, the CSE detection, and the person mask when producing the output image, thus turning the architecture into a conditional GAN. The encoder is modeled more or less as a "reversed" version of the StyleGAN decoder without the modulation of feature maps. Skip connections are added between the feature maps of corresponding resolutions for the encoder and decoder part of the generator in a U-net [46] fashion, such that the generator is able to better utilize the spatiality of the input content. The decoder part of the generator, as well as the discriminator, are identical to those used in StyleGAN2.

4.5 Conditional GAN Loss

This baseline generator is trained solely by feedback from the discriminator without any other loss terms. The GAN optimization objective is similar to that in section 2.10 with added conditional information, as follows:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D(x|B,M)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z|B,C,M)))]. \tag{4.1}$$

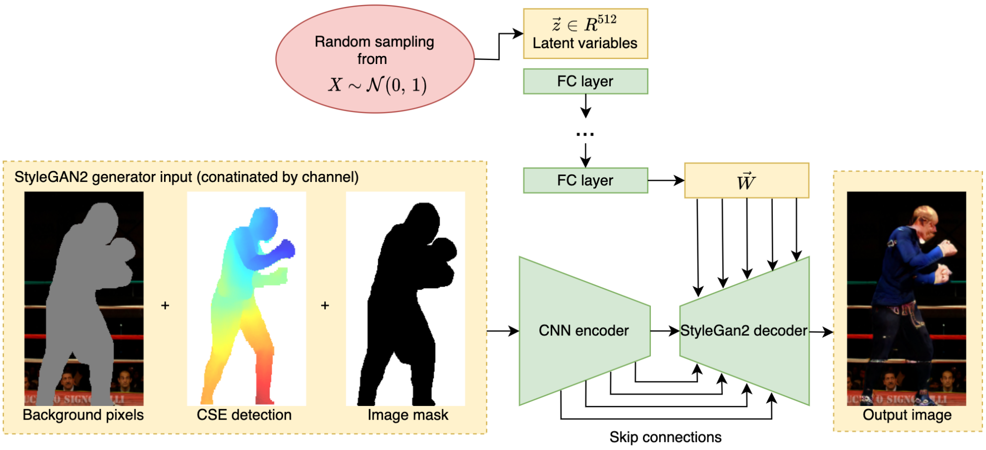


Figure 4.4: Baseline StyleGAN generator architecture. The model uses the background information, the person mask, and the CSE prediction as conditioning and tries to use this information to generate a realistic person fitting the background. The generator architecture follows the pattern of StyleGAN (subsection 2.10.5) with an added Unet encoder providing skip connection inputs to the standard StyleGAN decoder. The latent variable $\vec{z}$ is sampled from a gaussian distribution with a mean of 0 and a variance of 1.

Here, B is the background pixels, M is the mask and C is the CSE embedding values. G must now generate an image fitting B, C, and M, while D should check if the generated image fit with the given B and M.

4.6 Reconstruction for Better Consistency

The baseline model's main problem is that it relies heavily on the background and pose when deciding the appearance of the generated person. Ideally, the input to the encoder part of the generator should only supply the model with pose and context information, while the person's appearance should be given solely by the generated $\vec{z}$ vector. This does not happen with the baseline StyleGAN generator, where the person's appearance is often drastically altered with minor changes to the input background, pose, or person mask.

To create more consistent anonymizations, we propose to generate the latent vector $\vec{z}$ from the original person pixels and train the model to reconstruct the input. The reasoning behind this is to force the model to learn the mapping "appearance" $\rightarrow \vec{z} \rightarrow$ "appearance", with $\vec{z}$ representing all information not given as input to the generator encoder. To pull this reconstruction scheme off and still be able to sample new identities, we introduce two new terms to the loss functions of the model, namely discriminator Feature matching loss and Kullback-Leibler divergence loss.

4.6.1 Discriminator Feature Matching Loss

To train the model to reconstruct the input, we need a loss function that compares the original image to the generated image. Doing this comparison pixel-by-pixel is generally not a good idea, as looking at each pixel individually does not take structure into account and is generally not a good way of measuring image similarity [54]. A better option is to use features from a convolutional feature extractor when comparing the images. Using a ResNet pre-trained on ImageNet to create these features is possible. However, a model trained on ImageNet has not specifically been trained to separate different people from each other, which might restrain reconstruction performance. With this in mind, we decided instead to use the features coming from the discriminator for feature matching, as the discriminator, over time, will be able to create more and more useful representations for our dataset.

The goal of the feature matching loss is to make the features from the generated image be closer to the features of the original image, which is done in the following way: We denote the feature maps coming from the real and generated images as A_n and B_n, respectively, where n denotes the index of the feature map. Here, index 1 has the highest spatial resolution, index 2 has the next-to-highest spatial resolution, and so on. The loss value for one of the feature maps $\mathcal{L}_{\text{fm}}^n$ can be written as the absolute value of the difference between values of the two feature maps, as follows:

$$\mathcal{L}_{\text{fm}}^n = |A_n - B_n| \tag{4.2}$$

Note that n in this case denotes the index of the feature map. The process of computing $\mathcal{L}_{\text{fm}}^n$ for different feature map resolutions n is illustrated in Figure 4.5.

We use a weighted sum of the feature matching losses for each resolution to create the final feature matching loss, as expressed in Equation 4.3:

$$\mathcal{L}_{\text{fm}} = \sum_{n=1}^{N} \lambda_{\text{fm}}^n \cdot \mathcal{L}_{\text{fm}}^n \tag{4.3}$$

The weighting values λ_{fm}^n can technically be set to any real numbers. However, in our experiments, we have either set them to 1 or 0, effectively turning on or off feature matching for that specific resolution.

It is essential for the feature matching loss that the discriminator's weight values are "locked" and not updated while calculating the loss values. We do not update the weights as doing so would encourage the discriminator to generate

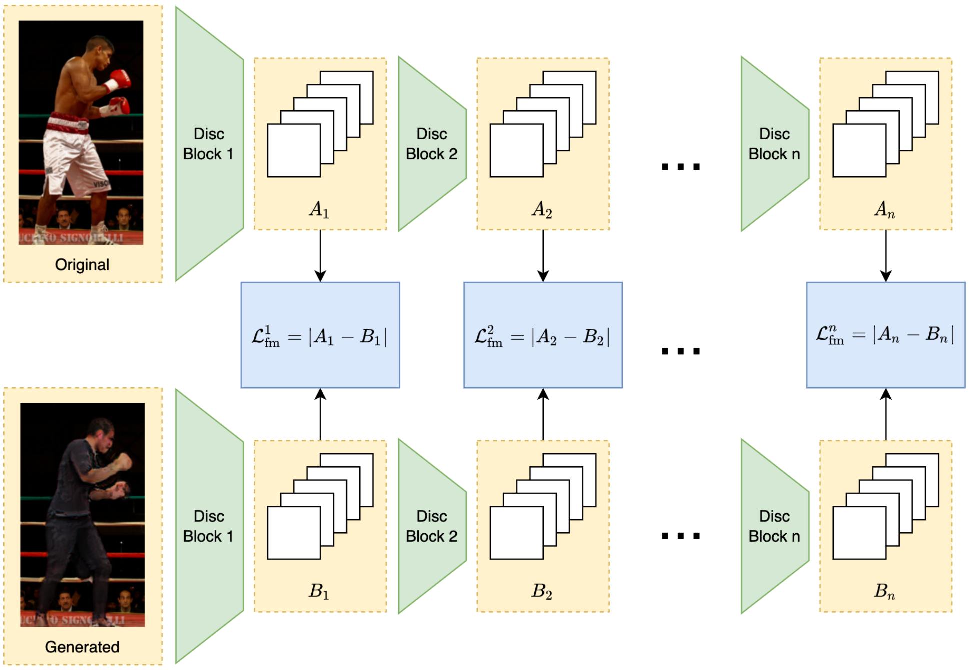

Original
Generated
Disc Block 1
Disc Block 2
Disc Block n
A_1
A_2
A_n
B_1
B_2
B_n
$\mathcal{L}^1_{\text{fm}} = |A_1 - B_1|$
$\mathcal{L}^2_{\text{fm}} = |A_2 - B_2|$
$\mathcal{L}^n_{\text{fm}} = |A_n - B_n|$
...

Figure 4.5: Illustration of how the discriminator feature matching loss is computed. Both the original image and the generated image are put through the same discrimanator, and a feature matching loss is computed for every feature map resolution.

similar features for the two images, which would be counterproductive. We do not want to tell the *discriminator* that these two images should be similar but rather to tell the *generator* that it should generate images that look similar in the eyes of the current version of the discriminator.

4.6.2 Kullback-Leibler Divergence Loss

The problem with introducing feature matching is that the distribution of $\vec{z}$ vectors will have no incentive to keep being Gaussian and can essentially end up having any possible distribution. Not knowing the distribution of $\vec{z}$ makes it problematic to sample random $\vec{z}$ values and might make the generator less able to generalize to new inputs because of the possible instability of $\vec{z}$. To counter this, we generate $\vec{z}$ much the same way as in a variational autoencoder (subsection 2.9.1) and add Kullback-Leibler (KL) Divergence loss.

The Kullback-Leibler Divergence [65] is a measure of the distance between two probability functions and is in the continuous case shown in Equation 4.4.

$$D_{kl}(P(z)||Q(z)) = \int P(z) \cdot \log\left(\frac{P(z)}{Q(z)}\right) dz \tag{4.4}$$

If we have that Q is given by a gaussian distribution $Q(z) = \mathcal{N}(0,1)$, that J is the dimensionality of z and that we assume $P(z)$ to be gaussian, the Kullback-Leibler Divergence between the two probability functions will follow by Equation 4.5 [26]:

$$D_{kl}(P(z)||Q(z)) = -\frac{1}{2}\sum_{j=1}^{J}(1 + \log(\sigma_j^2) - \mu_j^2 - \sigma_j^2) \tag{4.5}$$

Equation 4.5 is the KL Divergence loss term $\mathcal{L}_{\text{kl}}$ which we will use in our improved model. By introducing this loss, the distribution of the $\vec{z}$ vectors will fight between being turned into a Gaussian and providing reconstructive information to the rest of the generator.

4.6.3 Total Loss

The final loss function used for the model is a weighted addition of the standard GAN loss $\mathcal{L}_{\text{GAN}}$, the Feature Matching loss $\mathcal{L}_{\text{fm}}$ and the Kullback-Leibler loss $\mathcal{L}_{\text{kl}}$ shown in Equation 4.6:

$$\mathcal{L}_{\text{tot}} = \mathcal{L}_{\text{GAN}} + \lambda_{\text{fm}} \cdot \mathcal{L}_{\text{fm}} + \lambda_{\text{kl}} \cdot \mathcal{L}_{\text{kl}} \tag{4.6}$$

Throughout the thesis period we have experiment with different values for the weighting factors λ_{fm} and λ_{kl}.

4.7 Revised Generator Architecture With Apearance Mapper

The revised architecture using input pixels to generate $\vec{z}$ is shown in Figure 4.6. We have decided to call the mapping from input to $\vec{z}$ for an *appearance mapper*. The appearance mapper is modeled much the same way as the encoder part of a variational autoencoder.

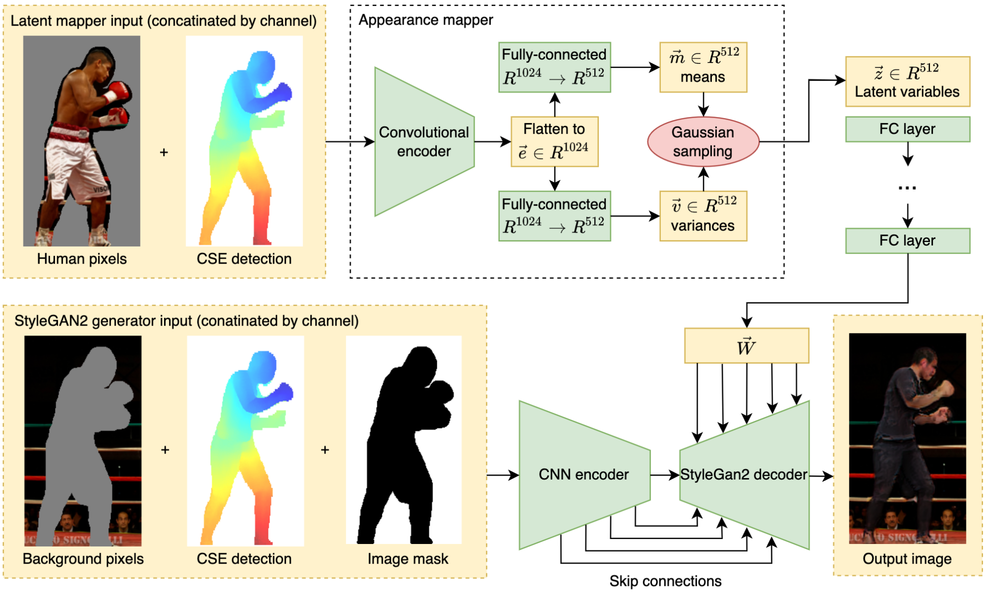

Latent mapper input (concatinated by channel)
Human pixels
+
CSE detection
Appearance mapper
Convolutional encoder
Fully-connected $R^{1024} \to R^{512}$
Flatten to $\vec{e} \in R^{1024}$
Fully-connected $R^{1024} \to R^{512}$
$\vec{m} \in R^{512}$ means
Gaussian sampling
$\vec{v} \in R^{512}$ variances
$\vec{z} \in R^{512}$ Latent variables
FC layer
...
FC layer
StyleGAN2 generator input (conatinated by channel)
Background pixels
+
CSE detection
+
Image mask
$\vec{W}$
CNN encoder
StyleGan2 decoder
Skip connections
Output image

Figure 4.6: Revised generator architecture from Figure 4.4 with added appearance mapper. The z-values are no longer drawn randomly but are instead created with the help of the original person's pixels in a process similar to the encoder part of a variational autoencoder. Adding the appearance mapper makes it possible to train the model reconstructively from input to output.

4.8 Normalizing the Feature Matching Loss

During experiments, it was found that the discriminator feature loss increased as time went on, even though the model was trying to minimize it. One of the reasons for this loss increase is probably that the magnitude of the discriminator features increases as the discriminator gets better at distinguishing fake images. To make the feature matching loss more stable, we tried to normalize the features of each feature map before calculating the feature matching loss. The normalization for each specific feature map was done by combining the statistics from the feature map of the real image with the statistics from the corresponding feature map of the generated image.

4.9 Co-Modulation

To let the input play a more prominent role in the generation and possibly model the context better, we experimented with implementing co-modulation [66] to the system. Co-modulation lets the modulation parameters be dependent on both the input and the $\vec{z}$ vector. To do this, a small separate network which we call a "Co-mod network" is added to the lowest resolution of the generator. This network creates separate modulation parameters, which are merged with the original modulation parameters through a single linear layer. This process is illustrated in Figure 4.7.

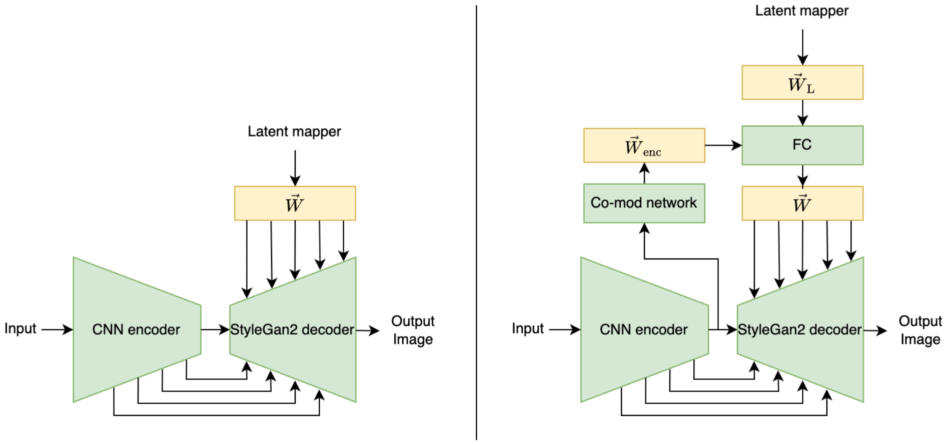


Figure 4.7: Co-modulation added to the generator of the revised architecture in Figure 4.6. **Left**: Original Generator. **Right**: Same generator with added co-modulation. With co-modulation, the modulation parameters are decided partly by $\vec{z}$ and partly by the input.

4.10 Improved Discriminator Conditioning

Throughout initial experiments, we found the model to be very sensitive to changes in the person mask when anonymizing, especially regarding hair and clothes such as shorts. We hypothesize that one reason for this sensitivity is that the *discriminator* focuses too much on the person mask when deciding if the images are real. For example, the discriminator might learn that a mask looking like it contains a pair of shorts will always come from a person wearing shorts. If the discriminator learns such a connection, it will probably penalize the generator if the generator tries to create a person wearing pants for a shorts mask. To make such connections harder to spot for the discriminator, we try to alter the discriminator conditioning. The new discriminator conditioning scheme is illustrated Figure 4.8 and consists of giving the CSE mask as input to the discriminator while filling the conditioning information with random noise so that the person mask is harder to pinpoint.

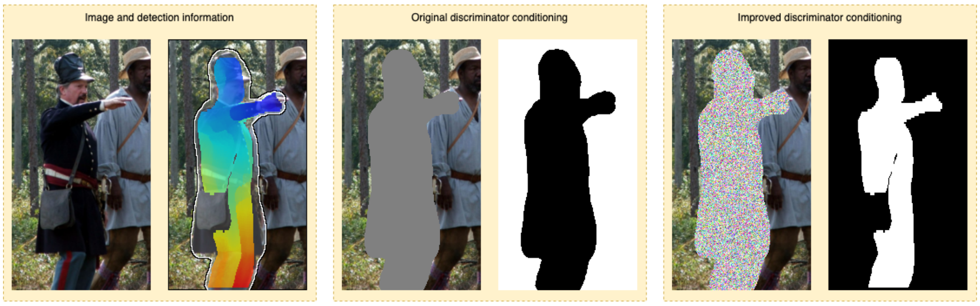


Image and detection information
Original discriminator conditioning
Improved discriminator conditioning

Figure 4.8: Improved discriminator conditioning. Instead of using the person mask, we send the embedding mask and a noise-filled version of the masked image. Notice how the jacket is not masked out for the embedding mask.

4.11 Comparison Metrics

In chapter 5, we will measure the quality of the generated images by using several metrics:

- **LPIPS reconstructed**: How much does the reconstructed image look like the original image?
- **LPIPS diversity**: How diverse are different anonymizations of the same image?
- **LPIPS**: How does a randomly generated image look compared to the original image?
- **FID**: How do the distributions of the real and generated images compare?
- **FID$_{\text{clip}}$**: An improved version of FID by using features from a CLIP model [53] instead of features from an ImageNet classifier.

The FID, FID$_{\text{clip}}$ and LPIPS metrics are described in subsection 2.13.2 and section 2.14. For LPIPS we will use the pre-trained w weights provided by the authors of the paper. The LPIPS diversity metric measures the diversity in the generated images by creating two random anonymizations for each validation image and checking the LPIPS distance between the two generated images, as proposed by Zhu et al. [47]. For the LPIPS diversity metric, higher is generally better, as it means the outputs are more diverse. LPIPS diversity should, however, not be the only metric to consider, as random noise, in general, will yield high diversity. We therefore need to look at the combination of the diversity score and FID, where we ideally want the combination of low FID and high diversity.

Chapter 5

Experiments and Results

This chapter will present our experiments and results. We will first state our experimental plan before describing how the experiments were carried out. Then, we will present qualitative and quantitative results for our final model by comparing it to the StyleGAN Baseline outlined in section 4.4. We will also present some qualitative results regarding our model's ability to do pose transfer and present an experiment where we check how anonymizing a person re-identification dataset degrades the data quality. Finally, we will do an ablation study of the various model improvements presented in chapter 4 and link to qualitative video examples.

5.1 Experimental Plan

We designed our experiments with the goal of improving the consistency of the baseline StyleGAN model in mind. The initial hypothesis was that the better the new model became at reconstruction, the better it would be at creating consistent anonymizations. Therefore, we primarily aimed at minimizing the LPIPS distance between the real and reconstructed images when deciding which experiments to run, starting with only reconstruction loss before adding the KL Divergence loss. After finding suitable weighting parameters for the loss functions, we performed ablations on all improvements in chapter 4, evaluating the models both quantitatively and qualitatively on images and video. Finally, we ran our best model and a comparatively large version of the StyleGAN baseline for an extended period to reach model convergence.

For the experimental results in section 5.3, we will provide both quantitative metrics and qualitative results. In addition, we will provide a video to better show how our model works in the video anonymization setting.

5.2 Experimental Setup

All experiments are run with the FDH dataset outlined in section 4.3 with 1 970 803 training images and 30 000 validation images. The dataset images are loaded using random horizontal flip as data augmentation with a flipping probability of 50%. For all the models, we use the Adam optimizer with a learning rate of 0.001 and the discriminator is regularized using r1 regularization [31] epsilon penalty [37]. We also use an exponential moving average of the generator as described in subsection 2.10.3.

All ablation models are run for 12 million training images, while the final model and the baseline StyleGAN model are run for 30 million images and 40 million images, respectively. Training is performed on the HEID cluster at the Department of Informatics at NTNU. This cluster contains 16 v100 GPUs with 32 GB of memory each. All ablation models are run on 1 GPU except for the larger model trained on 2, resulting in training times of around 2.5 days for these models. The StyleGAN baseline model is trained for 4.5 days on 4 GPUs, while the final model is trained on 2 GPUs for 7.5 days.

The metrics used for validating the images are the ones described in section 4.11. All metrics are computed exclusively on the validation images to measure how good the models are at generalizing to new, unseen images.

For more information regarding experiments, see the code attached to this thesis, containing all config files used in the thesis, complete with hyperparameters. The code is not provided online as it is based on some unreleased code written by my supervisor Håkon Hukkelås for the StyleGAN baseline. However, it might become available on GitHub at a later point in time. The code contains a readme so that it is easier to set up the project and find the config files used in the thesis.

5.2.1 Weighting of the Loss Functions

Through early experiments and some grid searching in the parameter space, we found that weighting the loss function values with λ_{fm} set to 5 and λ_{kl} to 0.01 yielded best results. All experiments in section 5.3 are run with these hyperparameters. The most important aspect of setting these parameters seems to be the relativeness of λ_{fm} to λ_{kl} and not their absolute values.

5.3 Experimental Results

The following subsections contain the results from all of our experiments in the thesis.

5.3.1 Comparison of Final Model to StyleGAN Baseline

Table 5.1 compares our final model to the StyleGAN baseline for our metrics, showing that our model has a drop in image quality and diversity compared to the StyleGAN baseline. Figure 5.1 and Figure 5.2 give image examples to show that our model better preserves the identity for changes in pose and background, respectively. Figure 5.3 shows some randomly generated anonymizations to showcase the diversity of the two models.

Table 5.1: Metrics for our model and StyleGAN Baseline. LPIPS Reconstructed is not included for the StyleGAN baseline because that model is not trained for reconstruction.

				LPIPS	LPIPS
	FID ↓	FID_{CLIP} ↓	LPIPS ↓	Reconstructed ↓	Diversity ↑
StyleGAN	**3.50**	**2.807**	0.2352	-	**0.1928**
Ours	5.16	2.971	**0.2152**	**0.1673**	0.1561

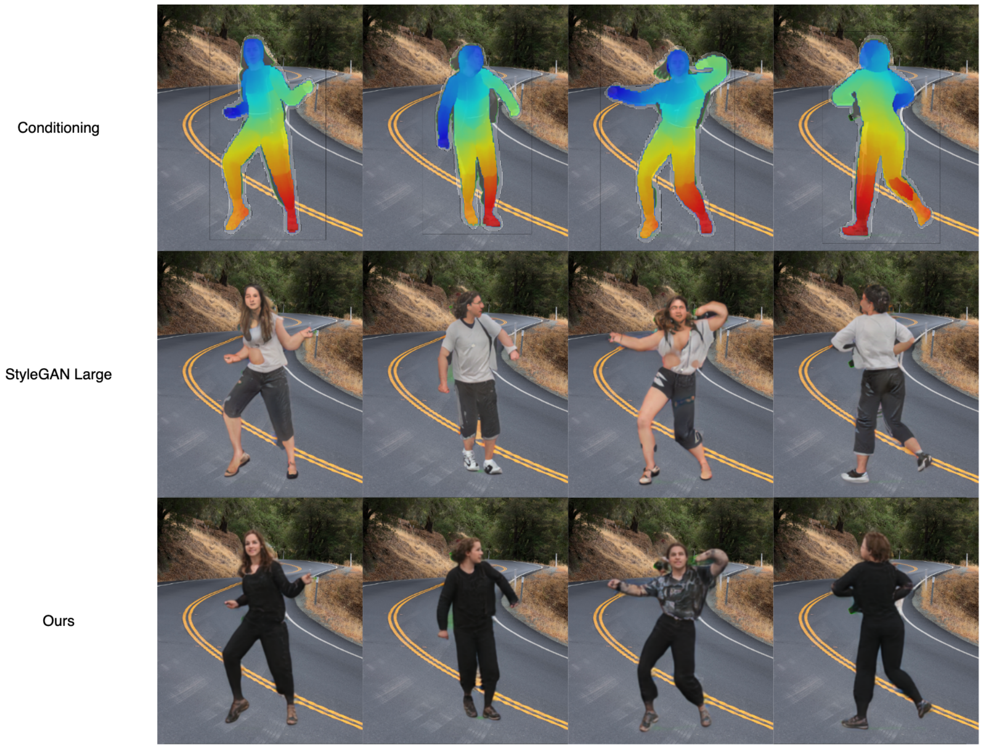


Figure 5.1: A comparison of how our model handles changes in poses compared to our StyleGAN baseline. The input images are photoshopped to have the same background but different pose. All anonymizations are done with maximum truncation (t=0)

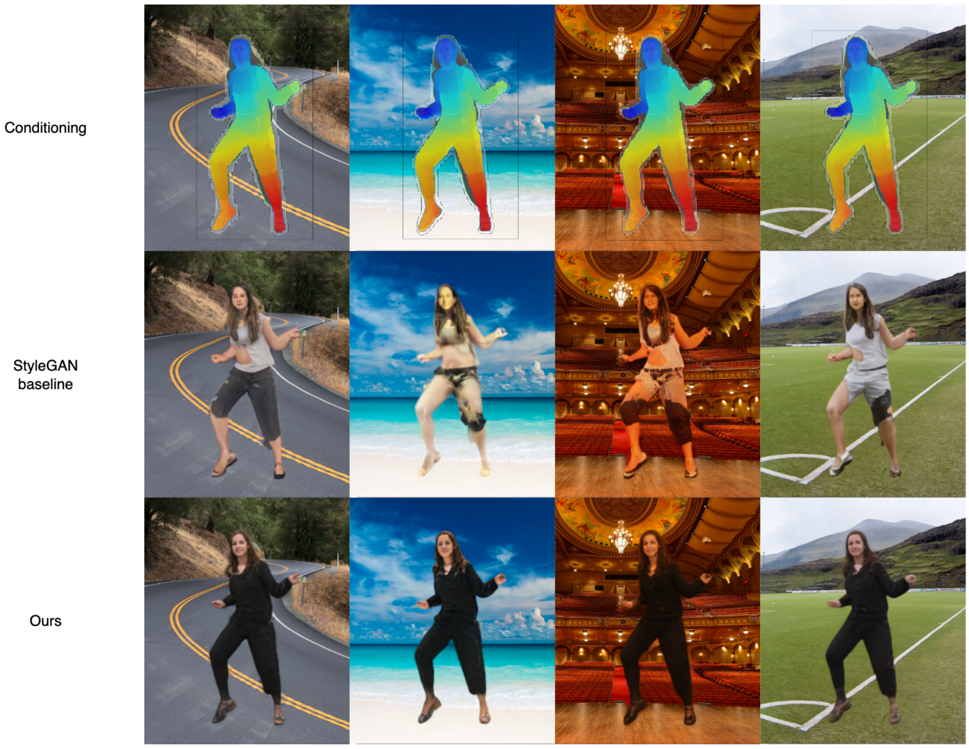


Figure 5.2: A comparison of how our model handles changes in backgrounds compared to our StyleGAN baseline. The input images are photoshopped to have the same pose but different background. All anonymizations are done with maximum truncation (t=0)

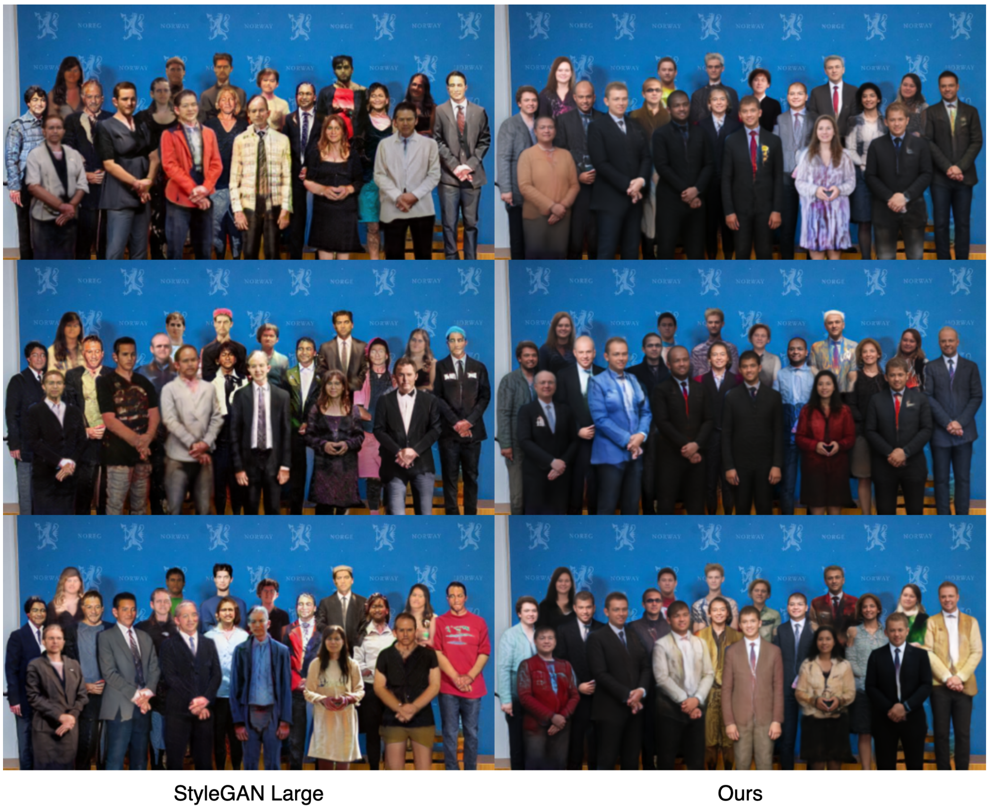


Figure 5.3: Diversity comparison between our model and StyleGAN baseline. All images are anonymized randomly with no trunctation (t=1). One can clearly see that the StyleGAN baseline provides more diverse anonymizations.

5.3.2 Anonyizing With an Input Image for $\vec{z}$

Figure 5.4 shows examples of our model reconstructing identities in new images by latent from the appearance mapper.

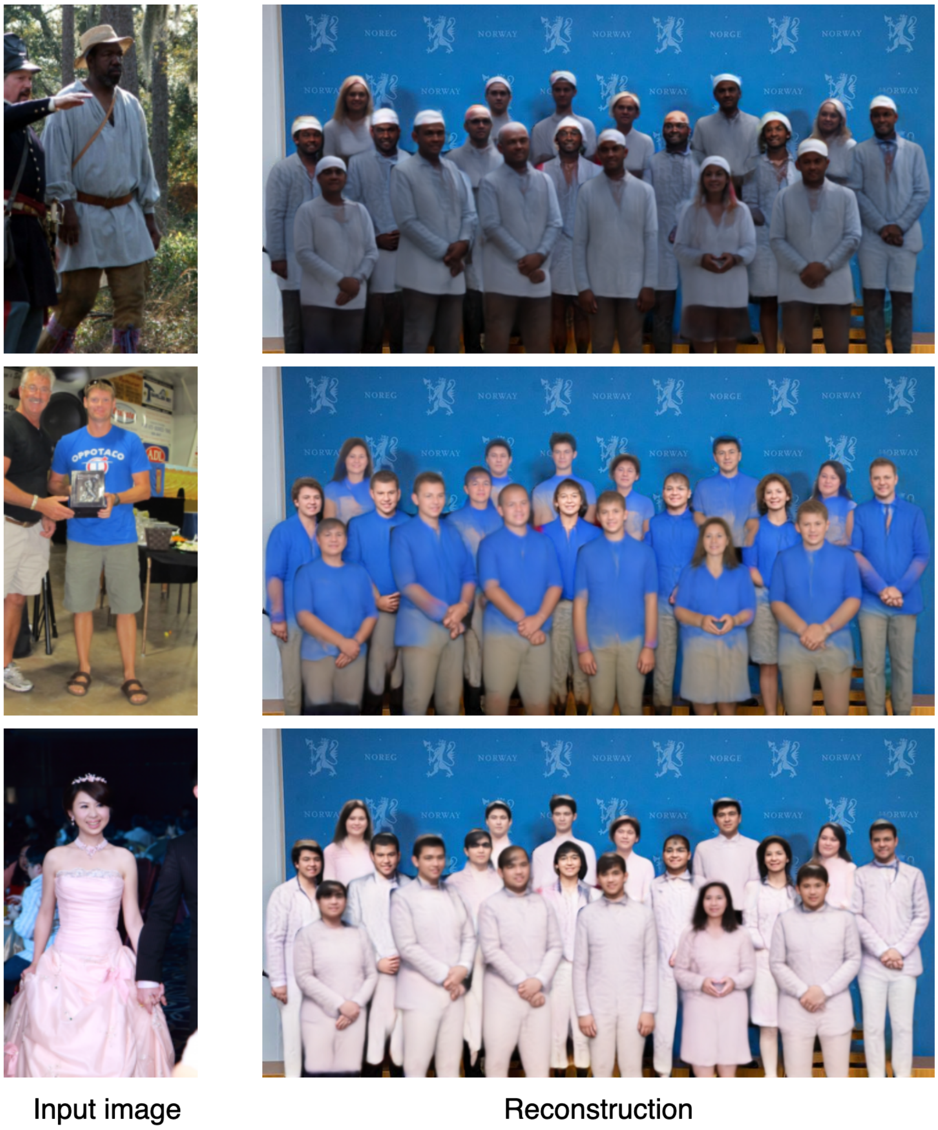


Figure 5.4: Reconstruction examples for the final model. The images on the left are used as input to the appearance mapper, and the output $\vec{z}$ vector is used to anonymize all persons in the corresponding image to the right.

5.3.3 Person Re-Identification

To assess the models' ability to preserve appearance across multiple poses and image contexts, we test how anonymizing a *person re-identifaction* dataset, namely Market1501 [9], affects the effectiveness of a person re-identification model. The anonymization is done by putting every image in the Market1501 dataset through our anonymization pipeline while using the same $\vec{z}$ value for images of the same identity. For validating the results, we use a pre-trained OS-net model [67] with the validation script provided in the Torchreid library [68]. Table 5.2 shows that our model outperforms the StyleGAN baseline in terms of retaining the identity across images. However, there is still a massive drop in performance compared to using the original dataset. Rank-k means that there is an identity match in the top k matched images, while mAP also takes the number of correctly matched images into account [9].

Table 5.2: Person re-identification results for anonymized versions of Market1501 compared to the original dataset. The validation is done using a pre-trained Osnet model.

	mAP ↑	Rank-1 ↑	Rank-5 ↑	Rank-20 ↑
StyleGAN Baseline	3.2	10.5%	18.7%	28.3%
Our Model	**14.5**	**41.0%**	**58.5%**	**71.6%**
Without anonymization	82.6	94.2%	97.9%	99.2%

5.3.4 Ablation on Model Improvements

The ablations are modeled as incremental experiments where one new component or improvement is added at a time. The first ablation model, "base", is the basic model outlined in section 4.7 with feature matching on resolutions 3 to 6 (which is the last resolution). The second ablation model, "+ more FM resolutions", changes so that the feature matching loss is run on resolutions 0 to 4, which we found to work better. The third model "+ co-modulation" adds co-modulation as described in section 4.9, while "+ impr disc conditioning" changes the discriminator conditioning to the scheme outlined in section 4.10. Finally, "+ larger model" is the same model trained with double the number of channels used in both the generator and the discriminator. The "StyleGAN Baseline" model is a StyleGAN model with comparable size to the four first ablations trained for an equal number of images.

Table 5.3 compares the various ablation models in terms of metrics, showing that all ablations except for "+ impr disc conditioning" increase image quality and reconstruction but decrease diversity. Figure 5.5 give image examples showing how the ablation models react to changes in pose, while Figure 5.6 similarly shows how the ablation models react to changes in background.

Table 5.3: Metrics for ablation models

				LPIPS	LPIPS
	FID ↓	FID_{CLIP} ↓	LPIPS ↓	Reconstructed ↓	Diversity ↑
StyleGAN Baseline	**4.12**	4.437	0.2461	-	**0.198**
Base	13.76	6.520	0.2464	0.197	0.170
+ more FM resolutions	10.25	6.492	0.2318	0.187	0.159
+ co-modulation	9.252	4.442	0.2274	0.185	0.149
+ impr disc conditioning	11.34	6.325	0.2273	0.187	0.141
+ larger model	7.569	**4.283**	**0.2177**	**0.178**	0.138

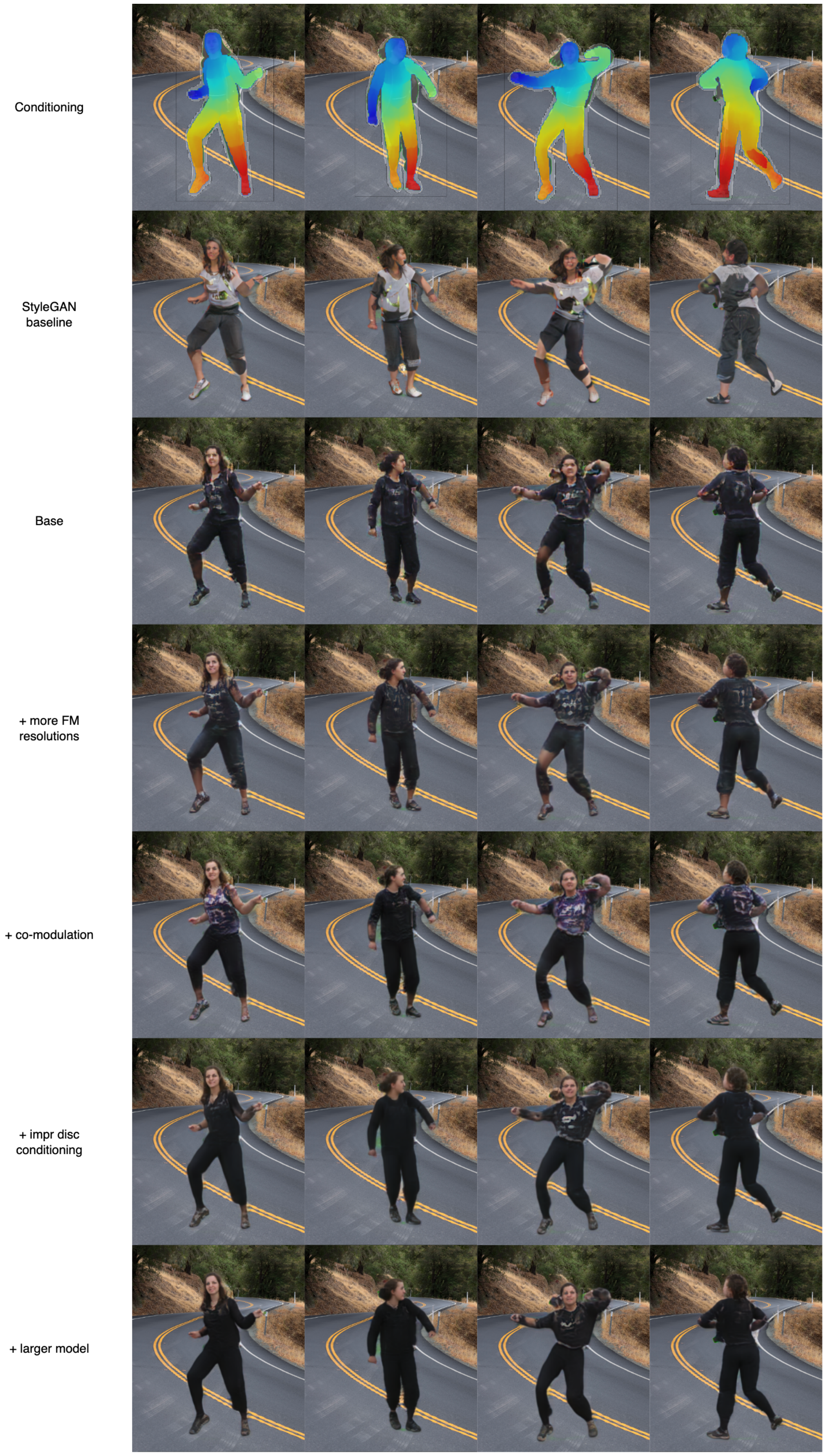


Figure 5.5: A comparison of how the ablation models handle changes in pose. All anonymizations are done with maximum truncation (t=0)

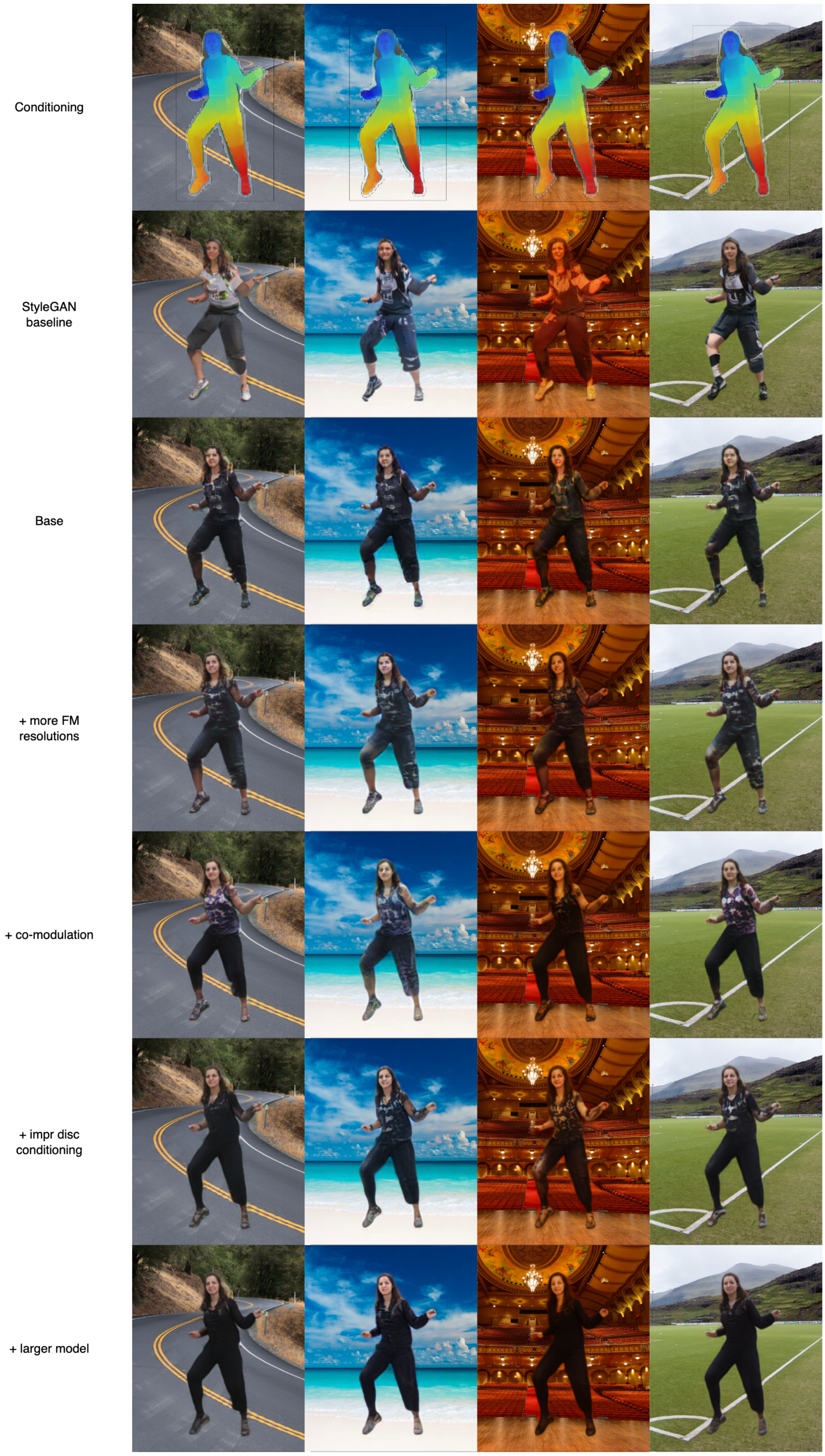


Figure 5.6: A comparison of how the ablation models handle changes in background. All anonymizations are done with maximum trucation (t=0)

5.3.5 Video Results

To give more insight into how the models compare for video anonymization, we have provided a link to an illustratory video in Figure 5.7. The video contains various qualitative results from anonymization on video. The results present in the video are:

- Visualized detection information on video
- The StyleGAN baseline visualized on video
- The final model visualized on video
- The StyleGAN baseline and final model on video with maximum truncation
- The StyleGAN baseline and final model on video for different backgrounds and poses
- Anonymization with $\vec{z}$ from input image on video for the final model
- A comparison of the ablation models with and without improved discriminator conditioning

For the videos marked with $t = 0$, we have used maximum truncation, meaning that the $\vec{z}$ is set to the 0-vector. Otherwise, tracking is used to maintain the same $\vec{z}$ for each person throughout the video. The persons who are only detected by Mask RCNN and not the CSE model are anonymized by applying heavy blur. All frames in all of the videoes are processed individually without taking the anonymized pixels in the adjecent frames into account. The last part of the video contains a comparison between the two ablation models “+ co-modulation” and “+ impr disc conditioning” to showcase the qualitative difference of adding the improved discriminator conditioning described in section 4.10.

Figure 5.7: QR code to video with more qualitative results. If you are viewing this on a digital device, you can also press this link: youtu.be0n_wob7CxwM.

Chapter 6

Discussion

The results in chapter 5 show clear improvements in consistency but also expose several problems regarding our optimization objective, the metrics, and the dataset. This chapter will discuss the results of chapter 5 in detail, describe the limitations of our model and approach, as well as pinpoint difficulties regarding the choice of architecture and dataset. We will also discuss further challenges related to consistent full-body anonymization and try to answer the research questions from section 1.2.

6.1 Disentanglement

The experiments in chapter 5 suggest that our model does better at disentangling pose, appearance, and context than the Baseline StyleGAN model. This can be seen from our person-reidentification results in Table 5.2, where we show that our model is better at generating persons with the same identity for varying images in the Market1501 dataset. We also show signs of increased disentanglement qualitatively in our pose and background examples of Figure 5.1 and Figure 5.2, as well as in the video from Figure 5.7. In these images and the video, we can see that our improved model is better at generating the same person when the background and pose change. However, this increase in consistency comes at a price of reduced image quality and diversity, which we can see in Table 5.1. As Table 5.1 suggests, the Baseline StyleGAN model outperforms our model in terms of image quality and diversity, meaning that the baseline model is probably the better choice if there is no need for consistency in the images to anonymize.

6.1.1 The Relation Between Appearance and Pose

A problem with the current pose representation is that it gives away quite some information regarding the original image subject. The CSE prediction contains clues regarding the person's body shape, while the person mask contains clues regarding traits such as clothes and hair. This appearance information present in the pose representation leads to biases in our model, such that it will only generate anonymizations of one gender for certain pose information. This effect can be seen in the video from Figure 5.7 and in the diverse results from Figure 5.3, where especially pose information with long hair results in women being generated. The pose information leaking appearance information is also a problem for clothes such as shorts, where some models will be more likely to generate bare legs if the mask seems to contain a pair of shorts.

Making an anonymization model where the generated appearance is totally unaffected by the original pose is a very challenging problem in general, as these two concepts are somewhat related. For example, generating a thin and long person where there once was a sumo wrestler would require filing in many background pixels. The other way around, generating a sumo wrestler where there once was a thin person would require filling in pixels outside of the person mask. For in-the-wild images, many such edge cases exist regarding pose, such as afros, long dresses, and hair blowing in the wind, to name a few. Examples of our model being unable to totally transfer the appearance of images can be seen for the reconstruction images in Figure 5.4. These images show that generating dresses or hats for the new pose information becomes a problem. Because of this intertwining of pose and appearance, generating the exact same person in every possible situation is really tricky and might not be a realistic goal at all.

6.2 Assessing the Ablation Improvements

In Table 5.3, we show that the ablation improvements made to the model give better results in terms of reconstruction, which we hypothesized to correlate well with consistency. The exception to this trend of better reconstruction performance in the ablations is the addition of the improved discriminator conditioning, which we introduced in section 4.10. From the metrics in Table 5.3, it might seem that adding the improved discriminator conditioning results in lower performance. However, we will argue that the qualitative results in Figure 5.5, Figure 5.6, and especially the video in Figure 5.7 suggest otherwise, at least in terms of disentanglement.

One of the problems is that we do not know if an increase in reconstruction performance happens because the model is better disentangled or if it is better at picking up hints from the pose and background information. The improved discriminator conditioning from section 4.10 was designed to make it harder for the model to pick up on pose hints, so it is not that weird that the resulting model has decreased reconstruction capabilities.

We originally designed the new architecture outlined in section 4.7 so that the $\vec{z}$ vector should be used when deciding on the generated person's appearance. However, nothing stops the model from also using the pose and background information when choosing appearance, which it also does in many cases, as described in subsection 6.1.1. Because of this, we argue that the reconstruction metric alone is not a sufficiently good indicator of disentanglement. This fact can also be seen for the models trained without KL loss, which are good at reconstruction but bad at generalizing to new poses, as will later be discussed in section 6.4. The lack of good metrics is a problem both for us when evaluating our models against each other and for future researchers wanting to improve on our work. Using person-reidentification degradation as an evaluation metric is a viable option but also has problems, which are discussed in the next section.

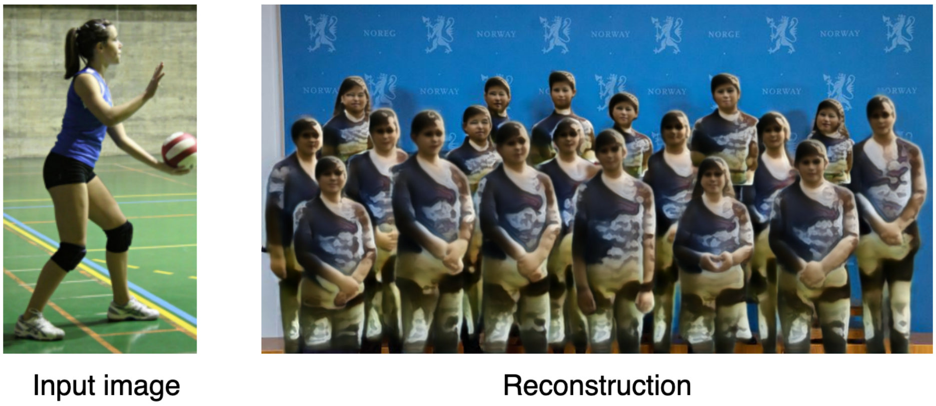
6.3 Issues with Person Re-Identification

In subsection 5.3.3, we show that our model degrades the quality of a person-reidentification dataset less than the StyleGAN baseline. However, there is still a massive drop in all metrics for both models compared to a dataset without anonymization. This huge performance drop is not only our model's fault but also comes from the fact that our detection system is not good enough on the Market1501 dataset. Our detection system could only create a CSE prediction for 72.4% of the images in Market1501. In addition, 22.5% of images did get a heavy blur on the predicted Mask RCNN mask, and 5.1% of the images had no detection whatsoever. The CSE detection rate of 72.4% means that a random pair of images drawn from an anonymized dataset only has a 52,4% chance (squared detection rate) of containing two images anonymized by our model. On the positive side, there is only a 0.26% chance that a random image pair contains two images that are untouched by our pipeline because of no detection. These issues show that our models' results in Table 5.2 are not comparable to the person re-identification result on the original dataset. However, the discussed issues are equal for both our final model and the StyleGAN baseline, meaning that it is reasonable to compare the models' performances to one another.

Several other aspects of our anonymization pipeline might also affect the performance of the pre-trained re-identification model. Some of the anonymizations might be based on CSE misdetections causing the generated images to look different from how they should. In addition, the current system has no guarantee that the identities generated are very different from each other. A random appearance vector $\vec{z}$ is created for each original identity, and some of these $\vec{z}$ might generate identities close to one another, meaning that using different $\vec{z}$ values might lead to different results. The problem of generating similar identities is especially the case for our model, which has decreased diversity in the generated images. Finally, anonymization might alter the context/domain of the images in the dataset since the anonymization model is trained on an image dataset with other properties. This change in context/domain might be unfamiliar to the pre-trained re-identification model. Training a person re-identification model from scratch might help alleviate this problem but would require a lot more time.

6.4 Reconstruction Loss is Not Enough

To illustrate that KL Divergence loss is necessary, we show a reconstruction example for a model trained without KL loss in Figure 6.1. As we can see, the output is severely corrupted. The reason for this is probably that the generator cannot tackle every possible $\vec{z}$ for every possible input pose and background, as the distribution of $\vec{z}$ can be highly irregular.



Input image

Reconstruction

Figure 6.1: Failure case for a model trained without KL Divergence loss. The image on the left is used as input to the appearance mapper, and the output $\vec{z}$ vector is used to anonymize all persons in the image to the right. The result does not resemble the input and contains severe artifacts. This effect seems to be more prominent when the change of pose is greater.

6.5 The λ Realism-Consistency Trade-Off

When introducing KL loss, setting the λ values for the loss terms defined in subsection 4.6.3 is quite difficult, as setting these values seems to be giving a trade-off between realistic anonymization and consistency. The model is quite sensitive to to the changes in the λ values, and as stated in subsection 5.2.1, the ratio of λ_{fm} to λ_{kl} seems to be more important than their absolute values.

Having a high ratio of reconstruction loss compared to KL loss increases reconstruction performance and seems to make the model more temporally consistent. However, increasing this ratio will also lower FID scores and eventually lead to artifacts and images not matching the context, as shown in section 6.4. If the reconstruction-to-KL ratio is small, however, the model will end up working much like our StyleGAN baseline, with an increase in image quality (FID) and lower reconstruction. Choosing how much realism should be prioritized over consistency is not straightforward and might depend on which trait one finds most important. The final parameters used in subsection 5.2.1 seem to give a decent trade-off between realism and consistency. However, we have not done very extensive grid searches in the hyperparameter space as our model takes much time and resources to train.

One of the reasons why large reconstruction loss leads to less realism seems to be that the GAN training is disturbed by the resulting heavy clash between KL Divergence loss and feature matching. When increasing the feature matching loss, the distribution of $\vec{z}$ will stray further away from a Gaussian, meaning that the KL-loss will increase to push the distribution of $\vec{z}$ back to a Gaussian. After a while, the KL-loss seems to stabilize around a certain value, and this value will represent more or less how Gaussian $\vec{z}$ is, with lower values meaning that it is more Gaussian. In Figure 6.2, we can see a correlation between high KL Divergence loss and large differences between real and fake scores for real and generated images. This large difference in real and fake scores is not healthy for GAN training. The reason for the larger difference in real and fake scores is probably that $\vec{z}$ is less Gaussian for higher KL loss, or that the feature matching loss is dominating the gradients from the discriminator.

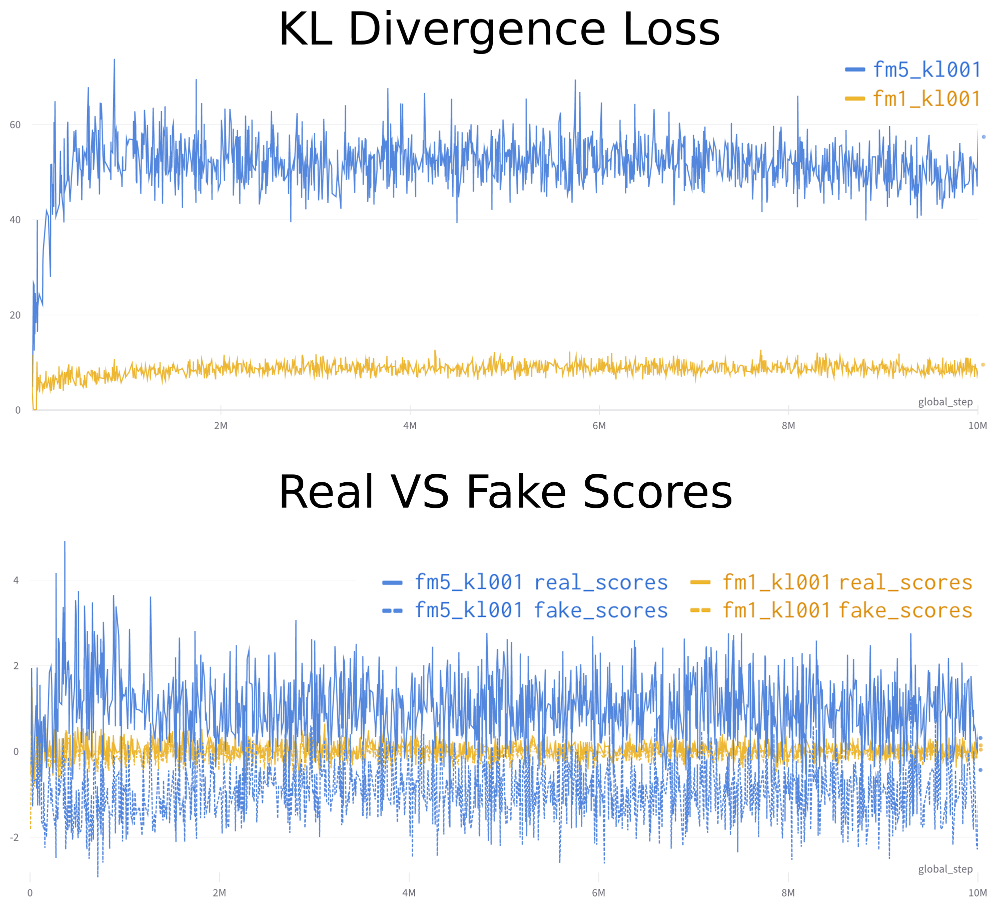


Figure 6.2: Signs of unhealthy GAN training for a model with high reconstruction loss (*blue*) and a model with lower reconstruction loss (*yellow*). The KL-divergence loss (*top*) is a lot higher when trained with more reconstruction loss, meaning that $\vec{z}$ is diverging from a gaussian distribution. The real and fake scores (*bottom*) are also further away from each other, meaning that training is more unhealthy.

6.6 Disentanglement Failures

The new model is able to disentangle pose, appearance, and context to some degree, as illustrated qualitatively in Figure 5.1, Figure 5.2, and the video in Figure 5.7. However, it is still quite a way to go, and our model cannot yet do very good video anonymization. Our model struggles especially when doing pose transfer of input images, such as in Figure 5.4. In many cases, it will prioritize creating more realistic results than accurately depicting the input individuals. One example where the model is not able to disentangle the features in the input image is shown in Figure 6.3. Here the model fails to reconstruct the shorts and bare arms of the input image and seems to use a lot of the input context when anonymizing. However, as opposed to the model trained without KL Loss in Figure 6.1, the generated people are more realistic.

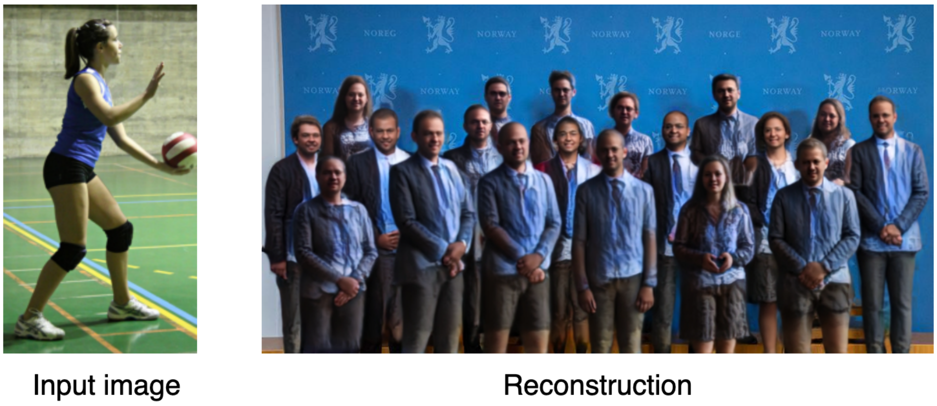


Figure 6.3: Failure in disentanglement for the final model when doing reconstruction. The image on the left is used as input to the appearance mapper, and the output $\vec{z}$ vector is used to anonymize all persons in the image to the right. The model fails to reconstruct the clothes of the input image and seems to use a lot of the input context when anonymizing.

6.7 CSE Failures

When working on real-world video, the Mask RCNN model seems to do a better job than the CSE model when it comes to detecting all individuals in the images. For the current anonymization pipeline, we get around this by applying a heavy blur to the Mask RCNN predictions without a corresponding CSE prediction, as seen in the video in Figure 5.7. This blurring is obviously not ideal, as we want to do as much realistic anonymization as possible.

The problems regarding CSE predictions can mostly be classified into these three categories:

- **No detections**: The CSE model does not create a prediction at all
- **Misdetections**: The CSE model creates a detection that does a bad job at describing the person
- **Temporal consistency in detections**: The detections vary a bit from frame to frame regarding the exact positioning of vertices.

Each of the problems for CSE described above propagates heavily into our anonymization pipeline and can easily be spotted when observing the output. We will propose several ways of handling these failures in chapter 7.

6.8 Research Questions

Below we will try to answer the research questions from section 1.2 based on what we have found during literature analysis, experiments, and discussion.

RQ1: What are the main challenges of consistent full-body anonymization?

Through dataset comparisons, we define the terms of *pose*, *appearance* and *context* in section 3.6, and in section 4.1 we state that a consistent full-body anonymization system should be able to disentangle these “factors-of-variation”. However, in the anonymization/inpainting setting, totally separating pose from appearance is difficult, as described in subsection 6.1.1. Based on this, an evident challenge regarding consistent full-body anonymization is to find out how, if at all, the model should be allowed to tailor the generated person to the given pose information. Allowing the model to use the pose information is somewhat problematic, as the pose (and mask) information can change significantly between image frames in a video or between different camera angles. However, there are many edge cases for in-the-wild images, such as dresses, shirts, long hair, and afros which make *not* taking pose information into account problematic.

In addition to the pose-appearance disentanglement issue, keeping the variance and the quality high in the generated images while at the same time improving consistency seems to be difficult, at least for our model with unpaired data. This can be seen for our ablation models in Table 5.3 and is also discussed in section 6.5.

RQ2: What datasets and pose estimation methods are suited for this task?

In section 4.2, we argue that there, for the current datasets described in subsection 3.6.1, is a trade-off between having paired images or diverse images. In this thesis, we have worked in the unpaired setting and used images with as much diversity as possible, which is essential if the system is to work for in-the-wild images. However, if one is able to create more diverse paired datasets or settle for systems working in limited environments, adapting full-body *synthesis* techniques to the task of full-body anonymization might surpass our proposed system in terms of disentanglement. Further possible work regarding paired data is described in subsection 7.2.1.

Regarding pose estimation, the current pose representation in FDH (section 4.3) does a very good job at describing the pose of the detected person, making our model's job easier. The predicted person masks in FDH better match the underlying person than the masks in COCO-Body, ensuring that our model sees less pixels of the underlying person such as hair, edges of clothes, and so on. However, we see a trade-off between accurate pose description of the anonymization subject and how much of the person's appearance (such as gender) can be inferred through the pose information. Other pose estimation methods and suggestions to further improve the disentanglement of our model will be discussed in subsection 7.2.4.

Another problem with the current pose estimation approach is that the CSE predictions are not always point, as described in section 6.7. This is bad for our video results in Figure 5.7 and also makes our person-reidentification results less accurate, as described in section 6.3. Possible improvements to the CSE detection system are discussed in subsection 7.2.3.

RQ3: How can we improve on existing anonymization techniques to make them more consistent?

In this thesis, we have, for the unpaired dataset setting, shown that improving on a conditional StyleGAN model by training it for reconstruction, as described in section 4.6, improves consistency. The increase in consistency is shown both by person re-identification results and qualitatively on images and video, as discussed in section 6.1. The results in Table 5.3 show that all improvements done to the model, with the exception of improved discriminator conditioning, result in improved metrics. However, we argue in section 6.2 that improved discriminator conditioning is, in fact, beneficial by referring to qualitative results, thus questioning our metrics for the experiments.

RQ4: How can we evaluate anonymization consistency?

Throughout the thesis period, we have found that quantitative evaluation of consistency (or disentanglement between pose, appearance, and context) through metrics is a difficult problem. Our original thought was that monitoring our model's reconstruction capabilities would be a good measure of consistency. However, we argue in section 6.2 that this is not always the case. We have also experimented with measuring how applying anonymization degrades a dataset's properties when it comes to person re-identification, as shown in subsection 5.3.3. However, measuring the degradation of re-identification datasets has several challenges. In our case, re-identification results after anonymization are not comparable to results on an unanonymized dataset due to the detection problems described in subsection 5.3.3. In addition, anonymizing a large person re-identification dataset like Market1501 takes much time (about 40 minutes for our model with cached detections), and the results will not tell that much regarding how the model is at using image context when anonymizing.

Trying to measure consistency qualitatively through images and videos is always an option, as we have tried to do in Figure 5.1, Figure 5.2, and subsection 5.3.5. However, qualitative evaluation takes much time both for generating the results and looking at them. In addition, the comparisons often become subjective, and comparing the results qualitatively between different research papers is problematic. We discuss possible further work regarding metrics better representing disentanglement performance in subsection 7.2.2.

Chapter 7

Conclusion and Further Work

This chapter will conclude the thesis and discuss further work regarding consistent full-body anonymization.

7.1 Conclusion

Creating systems able to generate consistent realistic anonymizations for persons in images will allow for more high-quality image and video datasets to be open to the public. In this thesis, we have introduced definitions of *pose*, *appearance*, and *context* in terms of consistent full-body anonymization and argued that a consistent full-body anonymization pipeline needs to disentangle these three "factors of variation". For the unpaired dataset setting, we have shown that extending a conditional StyleGAN model to try to make it learn the mapping "appearance" → $\vec{z}$ → "appearance" increases consistency. Our new model exhibits improved results qualitatively on images and video while outperforming the StyleGAN baseline when generating the same identity across images on the Market1501 dataset. However, the increase in consistency is a trade-off with image quality and diversity, and our model is still far from entirely disentangling shape, appearance, and pose. We argue through the report that creating a fully disentangled anonymization model is very difficult, as such a model should be able to generate any person in any context for every given pose, which is problematic for many real-world edge cases. Finding out how to handle the entanglement between pose and appearance present for subjects in real-world images is a real nut for consistent full-body anonymization.

7.2 Further Work

The task of consistent full-body anonymization is definitely not solved by this report, with several remaining challenges regarding datasets, metrics, pose estimation techniques, and model architectures. The following sections will describe some possible key areas which should be in focus for further research.

7.2.1 Paired Data

State-of-the-art for disentangled human synthesis methods use paired data, as described in chapter 3. Having paired data where it is possible to see the people to anonymize from different angles, poses, and backgrounds simplifies the task of creating a training pipeline for better disentanglement. The problem with paired data, however, is that it is considerably harder to gather. Moreover, when the data is harder to gather, it would probably also be less of it, meaning less variation as described in section 3.6. One solution to the variation problem might be to train the model on both paired and unpaired data by, for example, using 50 % of each type. It would also be possible to create artificial pairs from single images by applying data augmentations such as random horizontal flipping, random cropping and rotation, random white balance, and others.

One way of gathering paired data would be to use the same dataset generation pipeline described in subsection 4.3.2 but on video data. For example, one can use a tracking model on the input videos to gather identity mappings of the people present. Having video data (and not just pairs) might also open up the possibility of using video-generation techniques [69] which takes adjacent frames into account during generation. The YFCC100M dataset used to generate the FDH dataset contains some videoes, but these do probably not have the same variation present for the photos. In addition, there are also Multi-object tracking and segmentation (MOTS) datasets [70] which might be suitable in this setting.

7.2.2 Improved Metrics

No perfect metrics exist for evaluating consistent full-body anonymization, as discussed in section 6.2. Developing reliable metrics that better reflect how well the model performs consistent anonymization is an important avenue of further research to better evaluate and compare models. As stated in section 2.1, both tasks, datasets, models, and metrics are needed for progress in a subfield of computer vision. Right now, for consistent full-body anonymization, the metrics are insufficient.

In a sense, we want metrics able to tell how good we are at disentangling pose, appearance, and context. Perceptual Path Length (PPL) [34] is one such metric used for disentanglement but is primarily meant for unconditional GANs and does not fit our purpose that well. We are not *that* interested in how the generation is altered by changing $\vec{z}$, but rather in how changes to input condition and background alter the generation. Another option than PPL would be to have a small amount of paired data to compare reconstruction from one frame to another. This would work fine for reconstruction, but we also want a model able to create diverse and *new* people who are consistent across frames.

Ideally, we might want some kind of pose-independent similarity measure so that it would be possible to assess the similarity between two images with different poses. One way of creating such a measure would be to anonymize rendered SMPL models with various poses and check the l1 similarity for corresponding model locations for the two anonymizations. This might work to some degree but misses many edge cases such as clothes and hair and does not model how good the model is at representing the context.

7.2.3 Improved Surface Predictions

Our proposed anonymization pipeline consists of two stages: detecting people and generating new people. In this thesis, we have made much effort on the generation part, but not the detection part. Our detection pipeline is not that robust, as described in section 6.7, and we are currently only using pre-trained models for it. The detection pipeline is used both when generating the FDH dataset and during inference, so all efforts to improve the detection system will probably lead to improvements in both dataset quality and final model output.

One way of improving the CSE predictions might be to give an already predicted person mask as input to the CSE model. This way, it would be possible to force every Mask RCNN prediction into having a corresponding CSE prediction and eliminate the problem of having unmatched Mask RCNN predictions. In addition, having a person mask as guidance when predicting the pose and only having to predict one CSE embedding at a time might make the pose predictions more accurate. Finally, having such a CSE model would open up the possibility of adding pose information to datasets already having ground truth person instance segmentations.

In addition, in the current dataset, we only save the pixel-to-vertex correspondences for every person and not the raw CSE prediction. When doing generation, we paste the embedding value for each pixel-to-vertex correspondence directly into the image, meaning that there will be some clusters of pixels having the same embedding value. This also means that the dataset currently has a finite set of possible CSE embedding values and that the model might have learned to separate these values clearly from another. Our input pose conditioning from "continuous surface embeddings" might, in fact, be read by our model as discrete and not continuous. If we were to store the raw CSE predictions for the dataset, this might lead to better temporal consistency, and more diverse anonymizations as the model would not be that dependent on exact embedding input.

A final method of improving pose predictions might be utilizing synthetic data to create a better pose predictor. This is already done for DensePose by Yan et al. [71], but their solution might not be directly applicable to our problem. An advantage of synthetic data for dense pose prediction is that no manual annotation of model-to-image points is needed, as the rendering engine can compute all these. If one decides to use the SMPL model for generating synthetic data, the AMASS [72] dataset contains over 40 hours of SMPL animations, while datasets like SURREAL [73] contain a library of SMPL textures. By making pose predictions on animation, it might also be possible to add mechanisms to make the CSE detection model more robust in terms of temporal consistency between frames.

7.2.4 Facilitating For More Possible Anonymizations

As discussed in subsection 6.1.1, the model still relies quite a bit on the input pose and person mask information when choosing how to anonymize the person. This is bad as it reduces the possible diversity of humans the models can output, leaving some characteristics of the original person in the anonymization. Some possible improvements regarding pose and person mask information for better disentanglement are provided below.

Less Restrictive Pose Conditioning

One way of allowing for more possible anonymizations is to change to less restrictive pose information like sparse keypoints. Sparse keypoints will give less information regarding certain aspects of the person, like the width of the hips and the exact placement of various body parts. In addition, it might be easier to create such pose-information temporally consistent across video frames. Sparse keypoints will, however, lead to other challenges, such as ambiguity in which foot is in front of the other and the exact angle the person is viewed from.

Data Augmentation on Person Masks

Even though we did some steps to improve how much the model relies on the person mask information in section 4.10, it is still quite sensitive to masks containing characteristics like long hair, shorts, and t-shirts. One way of improving this is to implement data augmentation on the person masks so that the correlation between person mask and body becomes less apparent. This augmentation should probably be done to both real and generated images and can be implemented in several ways. One could, for example, do a morphological dilation with various structuring elements or try to expand the person mask to mimic long hair or clothes. However, a problem with this approach is that it is only possible to increase the area of the person mask, as decreasing it might leave parts of the person outside the mask.

7.2.5 Modelling the Context Information From Input

A problem with the current approach is that the context information in some cases might not be given entirely from the background pixels. One example of this is the concept of blur. If a portrait is taken at a considerable focal length and large aperture, the background will be blurred while the person in front is sharp. With the current approach, our model will have no way of knowing if the person should be blurred as the background or if it should be rendered sharp. This poses a problem if we want to do anonymization in image domains where parts of the image can be blurred. Another example where context is different from the background is when the subject and background are lit differently, such as when the subject is in the shade or on a stage. One solution to this problem is to use input person pixels when deciding on the context. However, to do this, one must be very certain that identity information is not leaked through this context information.

7.2.6 Outlier Removal with Self-Distillation

Mokady et al. [39] propose a self-distillation approach to find and possibly remove outliers in uncurated datasets. The process is done by training a generative model for reconstruction and removing the examples that the model is not able to reconstruct sufficiently. The resulting dataset will contain a distribution that is easier for the model to generate. This way of filtering has the advantage of being specifically tailored to the model, as it leverages the fact that the generator will be best at reconstructing the more common features in the dataset. Self-Distillation is probably very applicable to our case, as it can be used both to find outliers and images with misdetected pose information. The images with misdetected pose information are generally more difficult for our generator to reconstruct as the generator cannot rely that much on pose information when reconstructing these images.

7.2.7 Other Model Architectures

There are probably several possible directions for further development of the current model architecture. Recent work regarding generators equivariant to translation and rotation [74], as well as architectures based on Transformers [75, 76] might provide improvements over the existing system in terms of disentanglement. In addition, there exists the possibility of more explicit separation of appearance, pose, and context in the $\vec{z}$ space. This is especially the case if the model is to be trained on paired images, where it might be possible to enforce parts of the $\vec{z}$ space to be the same between pairs of images.

Bibliography

[1] D. Anguelov, C. Dulong, D. Filip, C. Frueh, S. Lafon, R. Lyon, A. Ogale, L. Vincent and J. Weaver, 'Google street view: Capturing the world at street level,' *IEEE Computer*, vol. 43, pp. 32–38, Jun. 2010. DOI: `10.1109/MC.2010.170`.

[2] I. J. Goodfellow, Y. Bengio and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016, `http://www.deeplearningbook.org`.

[3] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006, ISBN: 0387310738.

[4] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd. USA: Prentice Hall Press, 2009, ISBN: 0136042597.

[5] T. Huang, 'Computer vision: Evolution and promise,' 1996.

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li and L. Fei-Fei, 'Imagenet: A large-scale hierarchical image database,' in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255. DOI: `10.1109/CVPR.2009.5206848`.

[7] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár and C. L. Zitnick, 'Microsoft coco: Common objects in context,' in *European conference on computer vision*, Springer, 2014, pp. 740–755.

[8] Y. Ge, R. Zhang, L. Wu, X. Wang, X. Tang and P. Luo, 'A versatile benchmark for detection, pose estimation, segmentation and re-identification of clothing images,' *CVPR*, 2019.

[9] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang and Q. Tian, 'Scalable person re-identification: A benchmark,' in *Computer Vision, IEEE International Conference on*, 2015.

[10] I. Sobel and G. Feldman, 'A 3x3 isotropic gradient operator for image processing,' *a talk at the Stanford Artificial Project in*, pp. 271–272, 1968.

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

[12] Y. Zheng, C. Yang and A. Merkulov, 'Breast cancer screening using convolutional neural network and follow-up digital mammography,' in *Computational Imaging III*, International Society for Optics and Photonics, vol. 10669, 2018, p. 1 066 905.

[13] R. Girshick, J. Donahue, T. Darrell and J. Malik, 'Rich feature hierarchies for accurate object detection and semantic segmentation,' in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[14] J. R. Uijlings, K. E. Van De Sande, T. Gevers and A. W. Smeulders, 'Selective search for object recognition,' *International journal of computer vision*, vol. 104, no. 2, pp. 154–171, 2013.

[15] R. Girshick, 'Fast r-cnn,' in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[16] S. Ren, K. He, R. Girshick and J. Sun, 'Faster r-cnn: Towards real-time object detection with region proposal networks,' *Advances in neural information processing systems*, vol. 28, pp. 91–99, 2015.

[17] K. He, G. Gkioxari, P. Dollár and R. Girshick, 'Mask r-cnn,' in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[18] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo and R. Girshick, *Detectron2*, `https://github.com/facebookresearch/detectron2`, 2019.

[19] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin and A. A. Kalinin, 'Albumentations: Fast and flexible image augmentations,' *Information*, vol. 11, no. 2, 2020, ISSN: 2078-2489. DOI: `10.3390/info11020125`. [Online]. Available: `https://www.mdpi.com/2078-2489/11/2/125`.

[20] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll and M. J. Black, 'Smpl: A skinned multi-person linear model,' *ACM transactions on graphics (TOG)*, vol. 34, no. 6, pp. 1–16, 2015.

[21] R. A. Güler, N. Neverova and I. Kokkinos, 'Densepose: Dense human pose estimation in the wild,' in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[22] N. Neverova, D. Novotny, V. Khalidov, M. Szafraniec, P. Labatut and A. Vedaldi, *Continuous surface embeddings*, 2020. arXiv: `2011.12438 [cs.CV]`.

[23] M. Ovsjanikov, M. Ben-Chen, J. Solomon, A. Butscher and L. Guibas, 'Functional maps: A flexible representation of maps between shapes,' *ACM Transactions on Graphics (ToG)*, vol. 31, no. 4, pp. 1–11, 2012.

[24] R. M. Haralick, S. R. Sternberg and X. Zhuang, 'Image analysis using mathematical morphology,' *IEEE transactions on pattern analysis and machine intelligence*, no. 4, pp. 532–550, 1987.

[25] D. Bank, N. Koenigstein and R. Giryes, 'Autoencoders,' *arXiv preprint arXiv:2003.05991*, 2020.

[26] D. P. Kingma and M. Welling, ‘Auto-encoding variational bayes,’ *arXiv preprint arXiv:1312.6114*, 2013.

[27] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville and Y. Bengio, ‘Generative adversarial nets,’ *Advances in neural information processing systems*, vol. 27, 2014.

[28] S. Wang and H. Liu, ‘Deep learning for feature representation,’ in Apr. 2018, ISBN: 9781138744387. DOI: `10.1201/9781315181080-11`.

[29] M. Arjovsky, S. Chintala and L. Bottou, ‘Wasserstein generative adversarial networks,’ in *International conference on machine learning*, PMLR, 2017, pp. 214–223.

[30] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin and A. C. Courville, ‘Improved training of wasserstein gans,’ in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan and R. Garnett, Eds., vol. 30, Curran Associates, Inc., 2017. [Online]. Available: `https://proceedings.neurips.cc/paper/2017/file/892c3b1c6dccd52936e27cbd0ff683d6-Paper.pdf`.

[31] L. Mescheder, A. Geiger and S. Nowozin, ‘Which training methods for gans do actually converge?’ In *International conference on machine learning*, PMLR, 2018, pp. 3481–3490.

[32] Y. Yaz, C.-S. Foo, S. Winkler, K.-H. Yap, G. Piliouras, V. Chandrasekhar *et al.*, ‘The unusual effectiveness of averaging in gan training,’ in *International Conference on Learning Representations*, 2018.

[33] A. Radford, L. Metz and S. Chintala, *Unsupervised representation learning with deep convolutional generative adversarial networks*, 2016. arXiv: `1511.06434 [cs.LG]`.

[34] T. Karras, S. Laine and T. Aila, ‘A style-based generator architecture for generative adversarial networks,’ in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[35] X. Huang and S. Belongie, ‘Arbitrary style transfer in real-time with adaptive instance normalization,’ in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1501–1510.

[36] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen and T. Aila, ‘Analyzing and improving the image quality of stylegan,’ in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[37] T. Karras, T. Aila, S. Laine and J. Lehtinen, *Progressive growing of gans for improved quality, stability, and variation*, 2018. arXiv: `1710.10196 [cs.NE]`.

[38] A. Brock, J. Donahue and K. Simonyan, ‘Large scale gan training for high fidelity natural image synthesis,’ *arXiv preprint arXiv:1809.11096*, 2018.

[39] R. Mokady, M. Yarom, O. Tov, O. Lang, D. Cohen-Or, T. Dekel, M. Irani and I. Mosseri, ‘Self-distilled stylegan: Towards generation from internet photos,’ *arXiv preprint arXiv:2202.12211*, 2022.

[40] M. Mirza and S. Osindero, 'Conditional generative adversarial nets,' *arXiv preprint arXiv:1411.1784*, 2014.

[41] L. Deng, 'The mnist database of handwritten digit images for machine learning research [best of the web],' *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012. DOI: `10.1109/MSP.2012.2211477`.

[42] P. Isola, J.-Y. Zhu, T. Zhou and A. A. Efros, 'Image-to-image translation with conditional adversarial networks,' in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134.

[43] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell and A. A. Efros, 'Context encoders: Feature learning by inpainting,' in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2536–2544.

[44] J.-Y. Zhu, T. Park, P. Isola and A. A. Efros, 'Unpaired image-to-image translation using cycle-consistent adversarial networks,' in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2223–2232.

[45] R. Zhang, P. Isola and A. A. Efros, 'Colorful image colorization,' in *European conference on computer vision*, Springer, 2016, pp. 649–666.

[46] O. Ronneberger, P. Fischer and T. Brox, 'U-net: Convolutional networks for biomedical image segmentation,' in *International Conference on Medical image computing and computer-assisted intervention*, Springer, 2015, pp. 234–241.

[47] J.-Y. Zhu, R. Zhang, D. Pathak, T. Darrell, A. A. Efros, O. Wang and E. Shechtman, 'Toward multimodal image-to-image translation,' *Advances in neural information processing systems*, vol. 30, 2017.

[48] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen and X. Chen, 'Improved techniques for training gans,' in *Advances in Neural Information Processing Systems*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon and R. Garnett, Eds., vol. 29, Curran Associates, Inc., 2016. [Online]. Available: `https://proceedings.neurips.cc/paper/2016/file/8a3363abe792db2d8761d6403605aeb7` `Paper.pdf`.

[49] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, *Rethinking the inception architecture for computer vision*, 2015. arXiv: `1512.00567 [cs.CV]`.

[50] 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,' *Advances in neural information processing systems*, vol. 30, 2017.

[51] Z. Liu, P. Luo, X. Wang and X. Tang, 'Deep learning face attributes in the wild,' in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 3730–3738.

[52] T. Kynkäänniemi, T. Karras, M. Aittala, T. Aila and J. Lehtinen, 'The role of imagenet classes in fr\'echet inception distance,' *arXiv preprint arXiv:2203.06026*, 2022.

[53] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, 'Learning transferable visual models from natural language supervision,' in *International Conference on Machine Learning*, PMLR, 2021, pp. 8748–8763.

[54] R. Zhang, P. Isola, A. A. Efros, E. Shechtman and O. Wang, 'The unreasonable effectiveness of deep features as a perceptual metric,' in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 586–595.

[55] H. Hukkelås, R. Mester and F. Lindseth, 'Deepprivacy: A generative adversarial network for face anonymization,' in *International symposium on visual computing*, Springer, 2019, pp. 565–578.

[56] M. Maximov, I. Elezi and L. Leal-Taixé, 'Ciagan: Conditional identity anonymization generative adversarial networks,' in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5447–5456.

[57] H. Hukkelås, M. Smebye, R. Mester and F. Lindseth, *Realistic full-body anonymization with surface-guided gans*, 2022. arXiv: `2201.02193 [cs.CV]`.

[58] L. Ma, X. Jia, Q. Sun, B. Schiele, T. Tuytelaars and L. Van Gool, 'Pose guided person image generation,' *Advances in neural information processing systems*, vol. 30, 2017.

[59] N. Neverova, R. A. Guler and I. Kokkinos, 'Dense pose transfer,' in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 123–138.

[60] L. Ma, Q. Sun, S. Georgoulis, L. Van Gool, B. Schiele and M. Fritz, 'Disentangled person image generation,' in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 99–108.

[61] K. Sarkar, V. Golyanik, L. Liu and C. Theobalt, 'Style and pose control for image synthesis of humans from a single monocular view,' *arXiv preprint arXiv:2102.11263*, 2021.

[62] Z. Liu, P. Luo, S. Qiu, X. Wang and X. Tang, 'Deepfashion: Powering robust clothes recognition and retrieval with rich annotations,' in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[63] J. Johnson, A. Alahi and L. Fei-Fei, 'Perceptual losses for real-time style transfer and super-resolution,' in *European conference on computer vision*, Springer, 2016, pp. 694–711.

[64] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth and L.-J. Li, 'Yfcc100m: The new data in multimedia research,' *Communications of the ACM*, vol. 59, no. 2, pp. 64–73, 2016.

[65] S. Kullback and R. A. Leibler, 'On information and sufficiency,' *The annals of mathematical statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[66] S. Zhao, J. Cui, Y. Sheng, Y. Dong, X. Liang, E. I. Chang and Y. Xu, 'Large scale image completion via co-modulated generative adversarial networks,' *arXiv preprint arXiv:2103.10428*, 2021.

[67] K. Zhou, Y. Yang, A. Cavallaro and T. Xiang, 'Omni-scale feature learning for person re-identification,' in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 3702–3712.

[68] K. Zhou and T. Xiang, 'Torchreid: A library for deep learning person re-identification in pytorch,' *arXiv preprint arXiv:1910.10093*, 2019.

[69] N. Aldausari, A. Sowmya, N. Marcus and G. Mohammadi, 'Video generative adversarial networks: A review,' *ACM Computing Surveys (CSUR)*, vol. 55, no. 2, pp. 1–25, 2022.

[70] P. Voigtlaender, M. Krause, A. Osep, J. Luiten, B. B. G. Sekar, A. Geiger and B. Leibe, 'Mots: Multi-object tracking and segmentation,' in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7942–7951.

[71] H. Yan, J. Chen, X. Zhang, S. Zhang, N. Jiao, X. Liang and T. Zheng, 'Ultrapose: Synthesizing dense pose with 1 billion points by human-body decoupling 3d model,' in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10 891–10 900.

[72] N. Mahmood, N. Ghorbani, N. F. Troje, G. Pons-Moll and M. J. Black, 'AMASS: Archive of motion capture as surface shapes,' in *International Conference on Computer Vision*, Oct. 2019, pp. 5442–5451.

[73] G. Varol, J. Romero, X. Martin, N. Mahmood, M. J. Black, I. Laptev and C. Schmid, 'Learning from synthetic humans,' in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 109–117.

[74] T. Karras, M. Aittala, S. Laine, E. Härkönen, J. Hellsten, J. Lehtinen and T. Aila, 'Alias-free generative adversarial networks,' in *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.

[75] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, 'An image is worth 16x16 words: Transformers for image recognition at scale,' *arXiv preprint arXiv:2010.11929*, 2020.

[76] Y. Jiang, S. Chang and Z. Wang, *Transgan: Two pure transformers can make one strong gan, and that can scale up*, 2021. arXiv: 2102.07074 [cs.CV].

Appendix A

Filtering the FDH Dataset

When looking at images in the original FDH dataset, it was found that some images had bad CSE predictions. This was especially the case for images with weird body poses, challenging perspectives, only small parts of the body showing, and blur. Some examples of these misdetections can be seen in Figure A.1.

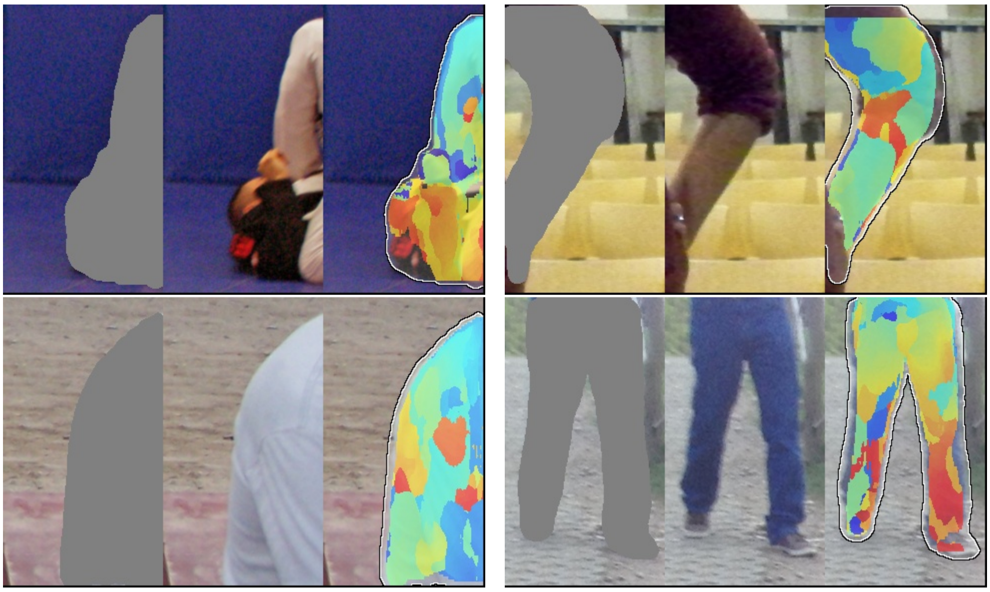


Figure A.1: Some images from the FDH dataset where the CSE predicitons have failed, creating pose information unsuited for good anonymization.

Most of the failed predictions were characterized by having few predicted vertices and often vertices predicted from all parts of the human at once. To try to filter out some of the misdetected images, we decided to look at the following filter criteria:

- How many unique vertices are present in the image
- How many unique vertices are there per body part (*e.g.* head, leg, etc.) in the image

A.1 Finding Body Parts From Vertices

To check for body parts in the image, we can use an existing mapping from vertices to body parts. An illustration of this mapping in "texture space" where each body part is color-coded can be found in Figure A.2.

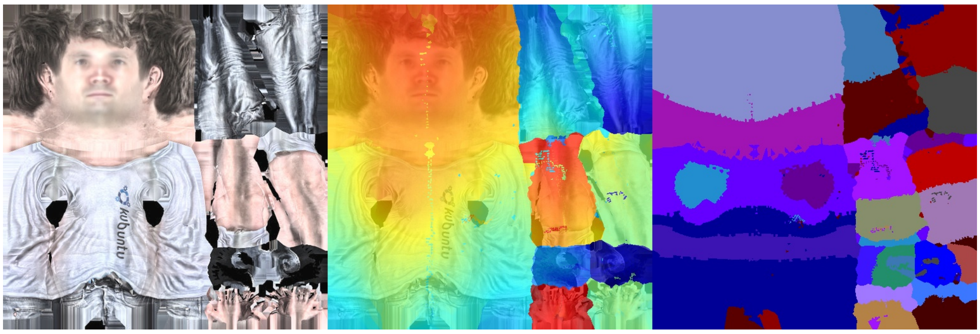


Figure A.2: An illustration of the locations of body parts in texture space.

By having this mapping between vertices and body parts, we can go from having a vertex value for each human pixel to having a body part value for each pixel. This makes it possible to find both the number of body parts present in the image and the mean amount of vertices per body part.

A.2 Comparing Filter Criterias for Finding Bad Detections

To check which filtering criteria are best for finding bad detections, it was necessary to go through at least some images manually. To make this process less tiresome, we decided to start by checking the filtering criteria on a sample of 2000 images from the FDH dataset. The images were sorted based on the filter criteria, and we looked at the bottom 100 images. Each of the images was placed into one of three categories:

1. Should be filtered away
2. Should probably be filtered away
3. Should be kept

We decided to have three categories instead of two because it is difficult to draw a clear line for what can be counted as a misdetection. For some images, it might depend a bit on the person deciding. Examples of images with borderline failed detections can be shown in Figure A.3.

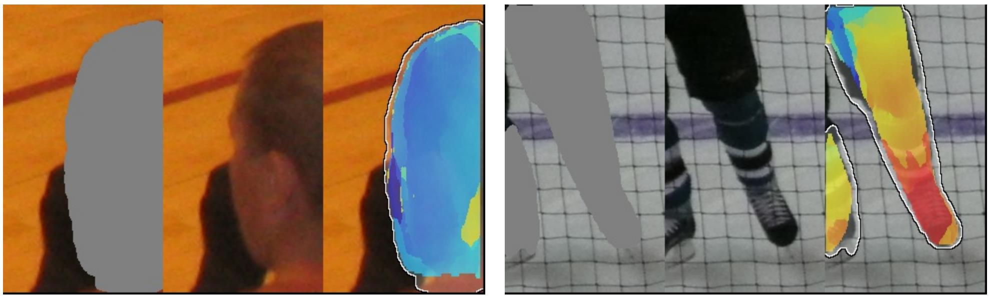


Figure A.3: Some images from the FDH dataset where it might be debatable whether the detection should count as a misdetection or not. Images like these are marked as "Should probably be filtered away"

The results from going through the 100 lowest-ranked images for both the filter criteria "amount of unique vertices" and "amount of unique vertices per body part" can be found in Table A.1.

Table A.1: Results from the two filter criterias on finding bad detections. The results are based on the 100 lowest ranked images on a sample of 2000 images from the FDH dataset.

	Num unique vertices	Num unique vertices per body part
Should be filtered away	34%	41%
Should be probably be filtered away	32%	36%
Should be kept	34%	23%

In Table A.1, we can see that using the number of unique vertices per body part gives a better distribution than just using the number of unique vertices. The reason for this is mainly that many of the failures consist of quite a bit of vertices but from different parts of the body. Some images filtered out by taking body parts into account but not by just using unique vertices can be found in Figure A.4.

A.3 Using Other Filter Criterias

It was discussed whether or not to filter the dataset for characteristics other than misdetections, for example, by the number of body parts and pixel standard deviation in the images. Filtering by pixel standard deviation made us find some quite challenging images in the dataset, as seen in Figure A.5. However, we decided

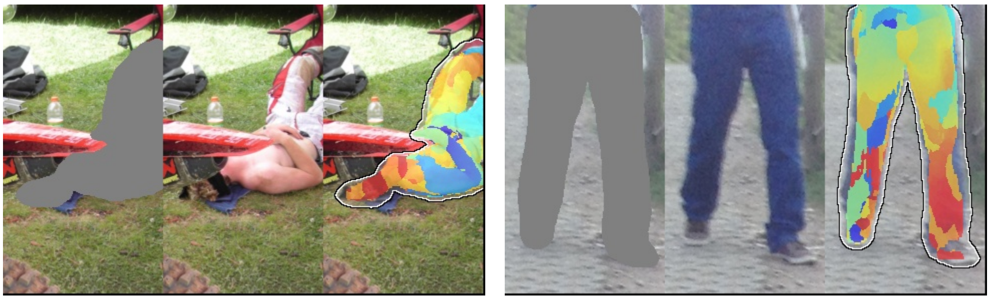


Figure A.4: Some example images which where not included in the bottom 100 images when sorting solely by unique vertices and were included when also taking body parts into account.

that filtering away anything other than misdetections would lead to less variation in the dataset, which we do not want.

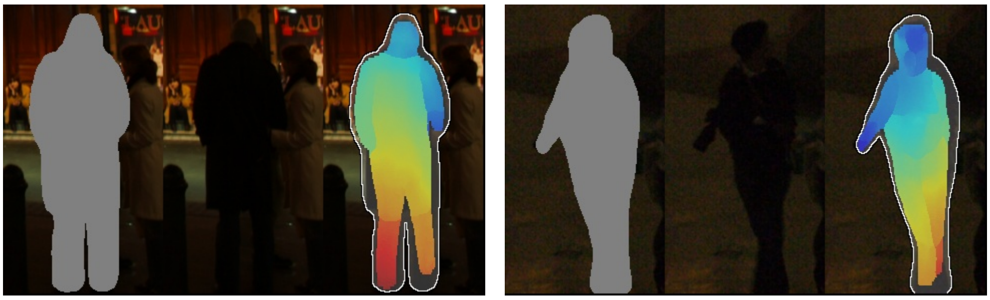


Figure A.5: Example of images in the dataset with low pixel standard deviation. We decided to not filter on this criteria to keep the variation in the dataset as large as possible

A.4 Choosing the Right Filter Value

After deciding to filter by "number of unique vertices per body part", finding a suitable threshold value was needed. To find the threshold value, we took the 2000 images sorted by filter value and classified the bottom 300 to check when the amount of misdetections started to decrease. We decided only to check the bottom 300 images as we did not believe it was necessary to filter out more than 15 % of the dataset.

The 300 images were classified using the same categories as in section A.2. Since the distribution of the images is quite spread out over the filter values, we plot the cumulative distribution of the images for each class against the filter value. These plots can be found in Figure A.6. Here we can see a clear trend that images with lower filter values are more likely to be misdetections.

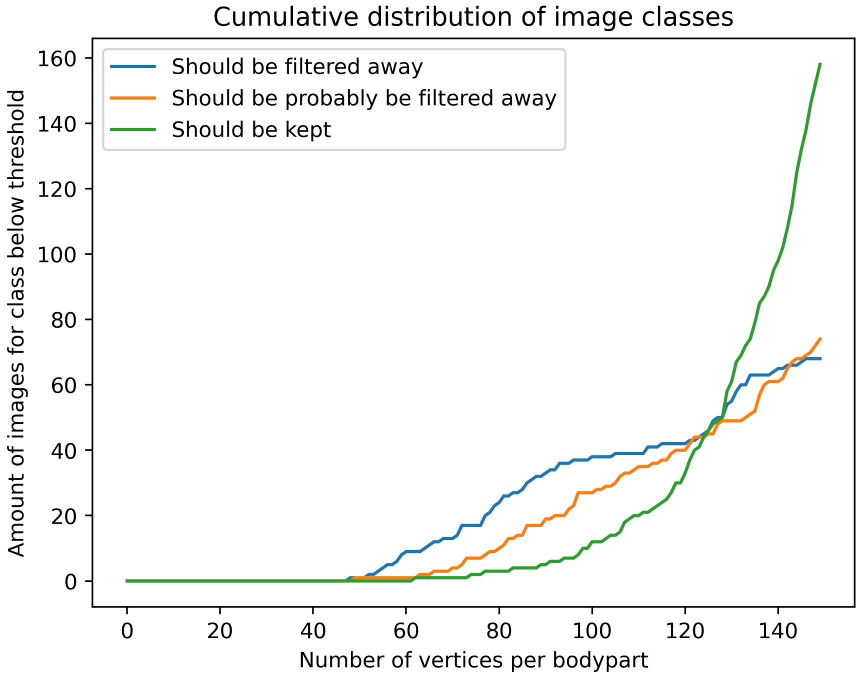


Figure A.6: The cumulative distribution of images with regards to "number of unique vertices per bodypart" for each category of the 300 manually categorized images from the 2000 FDH sample images.

Setting the final threshold based on the plots in Figure A.6 is not straightforward, but in the end, we decided to put the threshold at 135, which removes quite a bit of the misdetected images, but at the same time keeping more than 50 % of the good images. Setting the threshold at 135 means that we are removing about 10 % of the total images from the dataset. A 10 % data loss, in this case, is not that big of a problem, as we have many Flickr images to use.

A.5 Conclusion on Filtering

Filtering by setting an explicit thresholding value does a good job at removing misdetections but also removes quite a bit of usable images. If we had sufficient resources and time for manual sorting, a better approach would have been to set two thresholds and divide the images into three groups. Then it would be possible to remove all images from the first group, manually filter all images from the second group and keep all images from the last group. Manually filtering images is, however, far too much work for one person writing a master thesis.

Simen Holmestad

Towards Consistent Full-Body Anonymization

NTNU

Norwegian University of
Science and Technology