Håkon Hukkelås

DeepPrivacy: A GAN-based framework for image anonymization

Master Thesis, Spring 2019

Artificial Intelligence Group
Department of Computer and Information Science
Faculty of Information Technology, Mathematics and Electrical Engineering

TDT4900 - Computer Science, Master's Thesis
Main supervisor: Frank Lindseth
Co-supervisor: Rudolf Mester

Abstract

Freely collecting data from autonomous vehicles without anonymizing personal information is illegal with the introduction of the General Data Protection Regulation (GDPR). Therefore, to collect images to train and validate machine learning models, we are required to anonymize the data without drastically changing the original image appearance. Despite the remarkable progress of deep learning, there exists no suitable solution to automatically anonymize faces in images without destroying the original image.

We present *DeepPrivacy*; a two-stage pipeline that automatically detects and anonymize faces in images. We propose a novel generative model that can automatically anonymize faces in images while retaining the original data distribution; that is, our generative model generates a realistic face fitting the given situation. We ensure total anonymization of all individuals in an image by generating images without utilizing any privacy-sensitive information. Our model is based on a conditional generative adversarial network, generating images considering the original pose and image background. The conditional information enables us to generate highly realistic faces with a seamless transition between the generated face and the existing background. Furthermore, we introduce a diverse dataset of human faces, including unconventional poses, occluded faces, and a considerable variability in backgrounds. Finally, we present experimental results reflecting the ability of our model to anonymize images while preserving the data distribution, making the data suitable for further training of deep learning models. As far as we know, no other solution has been proposed that guarantees the anonymization of faces while generating realistic images.

Sammendrag

Samle data fra selvkjørende biler uten å anonymisere personlig informasjon er ulovlig etter introduksjonen av Personvernforordningen (GDPR) i 2018. For å samle data for å trene og validere maskinlæringsmodeller, må vi anonymisere dataen uten å endre det originale bilde betydelig. Selv med den store framgangen i dyp læring så finnes det ingen løsning som kan automatisk anonymisere fjes uten å ødelegge bildet.

Vi presenterer *DeepPrivacy*; en to-stegs modell som kan automatisk detektere og anonymisere fjes i bilder. Vi presenterer en ny generativ modell som anonymiserer fjes, samtidig som vi beholder den originale data distribusjonen; det vil si, vår generative modell genererer fjes som passer den gitte situasjonen. DeepPrivacy er basert på en betinget Generative Adversarial Network som generer bilder basert på plassering og bakgrunnen av det original fjeset. Videre introduserer vi et diverst datasett av menneskelige fjes, som inkluderer uvanlige rotasjoner av fjes, tildekket fjes, og en stor variasjon i bakgrunner. Til slutt, presenterer vi eksperimentelle resultater som reflekterer evnen til DeepPrivacy til å anonymisere fjes og beholde den originale data distribusjonen. Ettersom at de anonymiserte bildene beholder den originale data distribusjonen, gir det oss muligheten til å videre trene og validere maskinlærings modeller. Fra vår kunnskap finnes det ingen presentert løsning som garanterer anonymisering av bilder uten å ødelegge den original data distribusjonen.

Preface

In the following pages lies my master's thesis on how deep learning can anonymize data by generating new, unidentifiable faces. This master thesis is a part of a Msc. in computer science at the Norwegian University of Science and Technology, and this thesis is a part of the NTNU Autonomous Perception laboratory (NAPLab). I want to thank my supervisors, Frank Lindseth and Rudolf Mester, for excellent guidance and support through my project. The content of this master thesis is submitted as a 9-page paper to the British Machine Vision Conference (BMVC) 2019, and it is currently under review. The paper is named "DeepPrivacy: A generative adversarial network for face anonymization" and it is attached in the appendix.

Håkon Hukkelås

Trondheim, June 10, 2019

Contents

List of Figures

Glossary

AP Average Precision (AP) is a metric used for object detection, combining precision and recall to a single metric. 8

DSFD Dual Shot Face Detection (DSFD) is a SSD-based face detection method. 8

EMA Exponential Moving Average (EMA) is a technique to remove artifacts and improves the quality of images generated by GANs in inference. 20

EM-Distance Earth Mover's Distance (EM-distance) is a distance measure between two probability distributions. 16

FDF Flickr Diverse Faces (FDF) is a dataset consisting of 1.3M human faces with facial pose annotation and bounding box annotation of the face region. 5

FID Frèchet Inception Distance (FID) is a metric to evaluate the image quality of generated images compared to the original training data. 21, 22

Frèchet Distance Frèchet Distance is a measure of similarity of two curves that takes into account location and ordering of the points. 23

GAN Generative Adversarial Network (GAN) is a generative model that consists of a generator and a discriminator which are optimized together. 2

GDPR General Data Protection Regulation (GDPR) is a regulation in EU law on data protection and privacy for all citizens of the European Union and Europoean Economic Area. 1

IS Inception Score (IS) is a metric to automatically evaluate the image quality and sample diversity of generated images.. 21

JS-divergence Jensen-Shannon divergence (JS-divergence) is a distance measure between two probability distributions. 15

K-Lipschitz A function is a K-Lipschitz function if the function is lipschitz continuous for a real positive K. 16, 17, 52, 58

KL-divergence Kullback-Leibler divergence (KL-divergence) is a distance measure between two probability distributions. 13

Nash Equilibrium Nash Equilibrium is a proposed solution for a two-player non-cooperative game, where each player knows the optimal strategy; in other words, no player has anything to gain by changing their own strategy. 14, 52

non-maxiumum supression non-maximum supression is used to remove overlapping objec detections, making sure a single object is only identified once. 8, 32

recall Recall is a metric measuring the fration of relevant instances that have been retrieved over the total of relevant instances for a classification task. 3, 7, 31

RPN Region Proposal Network (RPN) is a anchor-based sliding window approach to generate regions of interest, often used for object detection. 8

S3FD Single Shot Scale-invariant Face Detector (S3FD) is an SSD-inspired face detection model using several layers to improve scale-invariant face detection. 37

SSD Single Shot Detection (SSD) is an object detection method that uses a single shot to detect multiple objects in an image. 8

VAE Variational Autoencoder (VAE) is a generative model based on an encoder-decoder structure to generate images. 13

WIDER-Face WIDER-Face is a face detection dataset consisting of a large diversity of human faces with different makeups, backgrounds, poses, illumination, and ethnicities.. 5, 8, 32, 39, 40, 41, 42, 43, 42, 43, 42, 44, 47, 49, 50, 57, 59, 71

Acronyms

AP Average Precision. *Glossary:* AP, 8, 32, 37, 43, 44, 57

DSFD Dual Shot Face Detection. *Glossary:* DSFD, 8, 32, 37, 43, 47, 50

EM-Distance Earth Mover's Distance. *Glossary:* EM-Distance, 16

EMA Exponential Moving Average. *Glossary:* EMA, 20, 40

FDF Flickr Diverse Faces. *Glossary:* FDF, 5, 31, 36, 37, 39, 40, 41, 42, 44, 45, 48, 49, 54, 57, 58

FID Frèchet Inception Distance. *Glossary:* FID, 21, 22, 23, 39, 41, 44, 48, 49, 52

FPS Frames Per Second. 7, 8, 11, 32, 37

GAN Generative Adversarial Network. *Glossary:* GAN, 2, 5, 13, 14, 13, 14, 15, 16, 17, 18, 19, 18, 19, 20, 23, 27, 28, 33, 36, 48, 51, 52, 59

GDPR General Data Protection Regulation. *Glossary:* GDPR, 1

IS Inception Score. *Glossary:* IS, 21, 22, 23

JS-divergence Jensen-Shannon divergence. *Glossary:* JS-divergence, 15, 16

KL-divergence Kullback-Leibler divergence. *Glossary:* KL-divergence, 13, 22

RPN Region Proposal Network. *Glossary:* RPN, 8, 10, 11

S3FD Single Shot Scale-invariant Face Detector. *Glossary:* S3FD, 37

SSD Single Shot Detection. *Glossary:* SSD, 8, 47

VAE Variational Autoencoder. *Glossary:* VAE, 13, 28

Chapter 1

Introduction

Privacy-preserving data-processing is becoming more critical every year; however, no suitable solution has been found to anonymize images without degrading the image quality or destroying the image. With the introduction of GDPR (General Data Protection Regulation), a suitable solution to successfully anonymize images is a necessity for a vast domain of applications, such as training and validation of deep learning models for autonomous vehicles. In this master thesis, we will evaluate a suite of possible solutions for anonymizing faces in images; yet, our approach is a general method, suitable for other applications such as anonymizing license plates.

1.1 Background and Motivation

The General Data Protection Regulation (GDPR) came to effect as of 25th of May, 2018, affecting all processing of personal data across Europe. GDPR requires regular consent from the individual for any use of their data, and the individual shall be able to withdraw their consent at any time. This law is difficult to obey for companies, especially in cases of training and validation of machine learning models. However, if the data does not allow to identify an individual, companies are free to use the data without consent.

Anonymizing personal data is a requirement before we store and process the data without consent. Furthermore, if we desire to train and validate deep learning models on the anonymized data, we require the anonymized data distribution to be similar to the original data distribution. In the case of faces in images, we want to replace the original face without destroying the existing data distribution; that is: the replaced face should be a realistic face fitting the situation.

Anonymizing images while retaining the original distribution is a challenging task. The model is required to remove all privacy-sensitive information, generate a highly realistic face, and the transition between the original and the anonymized parts has to be seamless. Standard tools, such as pixelation or blurring, are inadequate since they alter the original distribution: the output is not a realistic face. For practical use, we desire the model to be able to handle a broad diversity of images, poses, backgrounds, and different persons. Our proposed solution can successfully anonymize images in a large variety of cases, and create realistic faces to the given conditional information.

The majority of open-source datasets for autonomous vehicles are collected either in the United States or Germany. To apply models developed by the NAPLab in a Nordic climate, we wish to validate the models on data collected from similar environments. Collecting data for autonomous vehicles captures several privacy-sensitive parts, such as faces, and license plates. Before utilizing this data, we are required to anonymize it. This requirement is a challenging task to solve and is a major motivating factor for our work.

Generative Adversarial Networks (GANs)s have the potential to solve such a challenging task, and it has proven to be able to generate close to photo-realistic images of human faces. The original GAN introduced by Goodfellow *et al.* [14] was a proof-of-concept to model a data distribution; however, it was nowhere near to being applied in a practical application. With the numerous contributions since its conception, it has gone from a beautiful theoretical idea to a tool we can apply for real-world use cases. In our work, we show that GANs are an efficient tool to remove privacy-sensitive information without destroying the original data distribution.

1.2 Goals and Research Questions

The overall goal of this project is:

Goal *Develop a multi-stage pipeline to detect and anonymize faces in images, while retaining the original data distribution.*

Our goal is to anonymize all faces in images without destroying the existing data distribution; such that, it is still suitable to train and validate other machine learning models. The first stage of the pipeline should perform bounding box detection of faces with high recall; that is, it should be able to detect all privacy-sensitive parts in the image. From these detections, the second stage should generate a new face fitting the given situation, such that the transition between generated and real parts of the image is seamless.

Research Question 1 *What object detection and pose estimation methods are suitable for this application?*

Detecting privacy-sensitive information is the cornerstone of the entire framework. We require a robust model that can detect faces with high recall and exact bounding box annotations. Furthermore, the generative model might need additional information; for example, pose information, to successfully generate a realistic face.

Research Question 2 *How can we generate a realistic human face, fitting for a given situation?*

Anonymizing faces, while retaining the original data distribution, requires a complex generative model. To generate a face with a seamless transition to the original image, the generative model has to consider the existing background and pose of the original face.

Research Question 3 *How does the proposed framework perform on real images, which are not present in the training data?*

We train the final model on a dataset collected from a particular source, and the application area of this model can be different from the training data. For example, we might only use images collected in traffic; in this case, how will the model perform on anonymizing images from an airport?

Research Question 4 *How can we evaluate the impact of anonymized data used to train and validate deep learning models?*

By anonymizing our dataset, we will alter the original data distribution. This alteration might impact the dataset substantially, making it unsuitable for training or validation purposes. Quantifying the impact of anonymization is essential to evaluate the quality of our generated faces, and to assess the potentially degraded performance of a deep learning model. For example, we want to quantify how much the performance of a face detection model is degraded in the case of using an anonymized dataset rather than the original dataset.

1.3 Research Method

All research questions will be answered through experiments. Initially, we will review the existing techniques to detect and generate human faces, then select a technique to develop our model. Evaluating the result of the generated images is a challenging task, and we will assess it both quantitatively and qualitatively. For quantitative assessment, we will use a widely used performance metric to evaluate the image quality of our generated images. This quantitative measurement will give us a statistical metric of image quality, making it easy to evaluate architecture choices. However, these methods contain several disadvantages and results in an imperfect evaluation. Therefore, we present a wide range of anonymized images to qualitatively evaluate the perceived image quality.

A suitable dataset for this task has to contain faces with a high variance in poses, persons, and background clutter. To ensure that the model is generalizing well to unseen images, the model will be evaluated strictly on unseen images. To evaluate the impact of anonymization, we will evaluate our model on a different dataset than the training dataset, collected from a different source.

To ensure the reproducibility of our results, we will release the code with all hyperparameters for the final model. With our BMVC paper, we will release our proposed dataset (Flickr Diverse Faces), with all pre-processing stages described in Chapter 3. A final pre-trained model will be included with this report.

1.4 Contributions

We propose a model, called *DeepPrvacy*, that detects and anonymizes faces in images for a broad variety of applications. Our proposed model consists of an object and a human pose estimation model to detect bounding boxes and keypoints for the faces. This information is provided to a generative model to generate a new, anonymized face.

In summary, the main contributions of this thesis include:

- We propose a novel GAN architecture to anonymize faces, which ensures 100% removal of privacy-sensitive information in the original image. The generator can generate realistic looking faces that have a seamless transition to the existing background for various sets of poses and contexts.
- We provide the Flickr Diverse Faces (FDF) dataset, including 1.3M faces with a tight bounding box and keypoint annotation for each face. The dataset covers a considerably larger diversity of faces compared to previous datasets.
- We present an extensive qualitative and quantitative evaluation of the generated images and our model's ability to retain the original data distribution. Further, we perform several ablation experiments to illustrate the necessity of conditional pose information and a large generative model to generate realistic images.

1.5 Thesis Structure

In the following chapter, Chapter 2, we will give a brief overview of different object detection methods, generative models, and review closely related work. In Chapter 3, we present our model and discuss the architecture choices. Also, we present a new dataset, the Flickr Diverse Faces dataset. Chapter 4 presents the results of our model on the FDF dataset and the WIDER-Face dataset. In Chapter 5, we will discuss our research questions, limitations of our model, and perform an in-depth analysis of the model. Finally, in Chapter 6, we will conclude and discuss further work.

Chapter 2

Background Theory

In this chapter, we will cover the existing solutions for object detection and human pose estimation. Further, we will discuss several options for generative models, discuss techniques to evaluate generated images quantitatively, and take a deep dive into generative adversarial networks. Finally, we will review closely related work to the task of anonymizing visual data and why the existing solutions are inadequate to solve this challenging problem.

2.1 Object Detection and Pose Estimation

Object detection is the task of localizing and classifying objects in images. The task is a highly researched and fast moving area. For our task, we require a fast face detection method with high recall; that is, it is able to detect a majority of the privacy-sensitive faces in the image. Additionally, we desire an exact pose estimation of the face to further guide our generative model. In this section, we will shortly cover different face detection and pose estimation systems.

2.1.1 Face Detection

The challenge of detecting faces is an area of research that is well defined, and methods before the deep-learning era were able to solve this task well [79]. Traditional methods rely on hand-crafted feature vectors and a simple classification layer to generate proposals. For example, the Viola-Jones algorithm [69] can detect faces at 15 Frames Per Second (FPS), while still maintaining high detection accuracy. Recent work in this area has shifted its attention to detecting faces "in the wild," including challenging scenarios with high occlusion, reflection, image distortion, and low resolution faces, as small as 3px tall.

Current state-of-the-art face detectors can be roughly divided into two categories [35]; methods based on Region Proposal Network (RPN) introduced in Faster R-CNN [54]. Methods utilizing RPN are trained end-to-end and generate high-quality bounding box proposals, which are further refined by a regression head. The other category is Single Shot Detection (SSD) [41], which directly predicts bounding boxes and has no refinement stage. SSD methods have remarkably faster inference time and have recently attracted more attention.

TinyFace [22] is a RPN-based face detection model that significantly improved the state-of-the-art on the WIDER-Face dataset. Their results reduced the error rate on the WIDER-Face [75] "hard" by a factor of 2 from the previous state-of-the-art (from 29-64% Average Precision (AP) to 82%). They utilize a coarse image pyramid of the original image at different scales, then train separate detectors for the different scales. They merge the heatmaps from the image pyramid and apply non-maxiumum supression to generate the final heatmap. The detector is based on ResNet101 [20] and runs at 3.1 FPS on 720p resolution.

Since then, SSD based face extractors have become increasingly popular because of their superior inference time [35, 48, 82]. As of May 2019, the current state-of-the-art on both the WIDER-Face [75] and FDDB [25] datasets is the Dual Shot Face Detection (DSFD) [35]. Examples of DSFD impressive face detection can be seen in Figure 2.1. DSFD predicts on features extracted from several layers, similar to Feature Pyramid Networks [37]; however, they propose a feature enhancement module that enhances the discriminability and robustness of the features. Further, they perform slight adjustments to the loss function and improve the initialization of the anchor matching strategy to enhance the regressor. DSFD can perform inference in 22 FPS for VGA resolution on an NVIDIA P40 [35]. In comparison to the TinyFace [22] detector, TinyFace achieves 82% AP on WIDER-Face "hard", while DSFD achieves 90.4%.

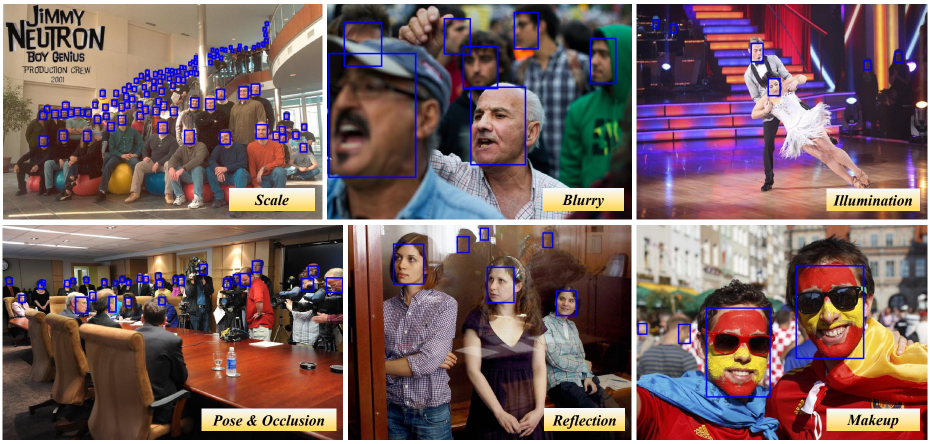


Figure 2.1: **DSFD Detection Examples:** Dual Shot Face Detection (DSFD) [35] is the current state-of-the-art face detection method. DSFD is robust to a large variety of scales, illumination, pose, reflection, and makeup.

2.1.2 Human Pose Estimation

Human pose estimation refers to the technique of detecting human poses in images or videos. The task is to predict a set of keypoints on the human figure placed on the different joints through the body. Pose estimation is often coupled with object detection and masking for improved results through multi-task learning [19]. OpenPose, Mask R-CNN, and DensePose are highly successful frameworks to perform both pose estimation and object detection.

OpenPose

OpenPose was the first real-time multi-person system to jointly detect a human body, hand, facial, and foot keypoints (in total 135 keypoints) on a single frame. The OpenPose framework consists of several techniques combined into a single keypoint detection framework and described in several papers from Carnegie Mellon University [10, 9, 62, 73]. The great benefit of this framework is its bottom-up approach for keypoint detection, giving a runtime that is independent of the number of persons in the image. The OpenPose system significantly exceeded the previous state-of-the-art on the MPII Multi-Person benchmark [4] and placed first in the COCO [38] 2016 keypoints challenge, and they continuously

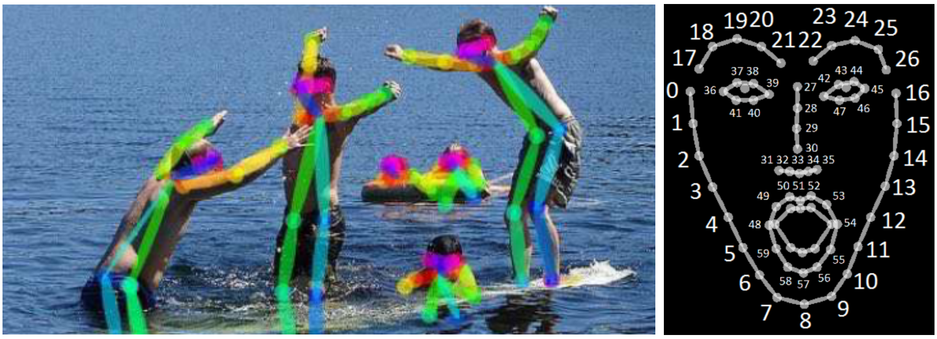


Figure 2.2: **OpenPose Predictions** *Left:* Examples of predicted poses from OpenPose. *Right:* Face keypoints that OpenPose predicts. Figure source: https://github.com/CMU-Perceptual-Computing-Lab/openpose

develop it. Figure 2.2 shows examples from OpenPose.

The significant advance of OpenPose is its bottom-up approach for keypoint detection. A common approach to keypoint detection is to employ a top-down system, first detecting persons in the image, then based on these detections perform single-person pose estimation. This approach suffers from being unable to recover in case the person detector fails, and the runtime is proportional to the number of persons detected in the image. The OpenPose framework employs a bottom-up approach that is separated into a two-branch pipeline that uses an iterative procedure to refine the prediction over successive stages. They utilize a VGG19 [63] network, specifically the first 10 layers, to generate a set of feature maps which is then fed into the first stage of each branch. Then the prediction is refined over t stages. By using a two-branch network, they can simultaneously predict body parts and the association between body parts.

Mask R-CNN

Mask R-CNN [19] is a flexible and general framework for object detection and instance segmentation. It provides a simple, yet efficient method to perform instance segmentation and its general framework is easy to extend to other tasks, such as human pose estimation. With this approach, they show state-of-the-art results on all three tracks of the COCO [38] suite of challenges, including object detection, instance segmentation, and person keypoint detection.

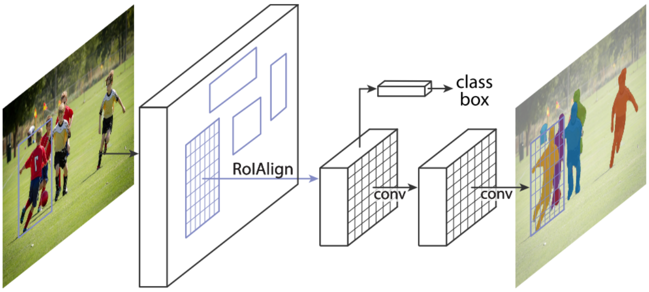


Figure 2.3: **The Mask R-CNN architecture** for instance segmentation. A deep CNN (e.g., ResNet/ResNext, VGG) extracts a set of feature maps; then region proposals are computed by the RPN. Finally, the region proposals are reshaped with RoIAlign and classified with outputs heads (in this case a classification head and instance segmentation head). Figure source: He *et al.* [19]

The Mask R-CNN framework improves the Faster R-CNN framework [54] by extending it to perform instance segmentation in addition to bounding box regression and object classification. Their main contribution is the replacement of the Region of Interest Pool (RoIPool) layer with a Region of Interest Align (RoIAlign) layer. This replacement is necessary as the RoIPool layer introduces a misalignment between the region of interest and the extracted features, which has a substantial negative effect on pixel-accurate instance segmentation masks.

One of their experiments to prove that Mask R-CNN is a general and straightforward framework for several tasks, was to add a new output head to perform human pose estimation. By modeling each keypoint on the body as a one-hot mask, they can output K one-hot masks for K keypoints. This is a method exploiting minimal domain knowledge, and they present marginally improved results from the COCO keypoint detection result of OpenPose. In comparison to OpenPose, the OpenPose network is a complex multi-stage pipeline, while Mask R-CNN is a more straightforward approach; however, the runtime of OpenPose is independent on the number of people in the image, while the RPN makes Mask R-CNN's inference time dependent on the number of persons.

DensePose

DensePose-RCNN [2] is a combination of two architectures: the Dense Regression [3] system and the Mask-RCNN [19] architecture. They state the problem

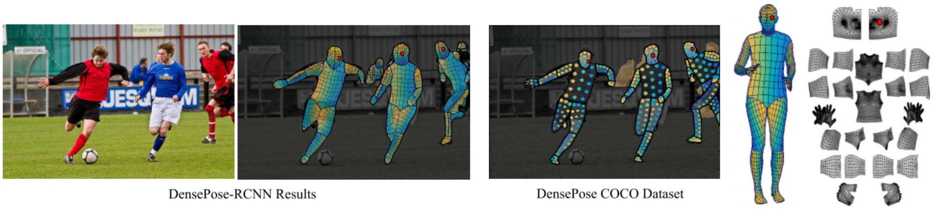
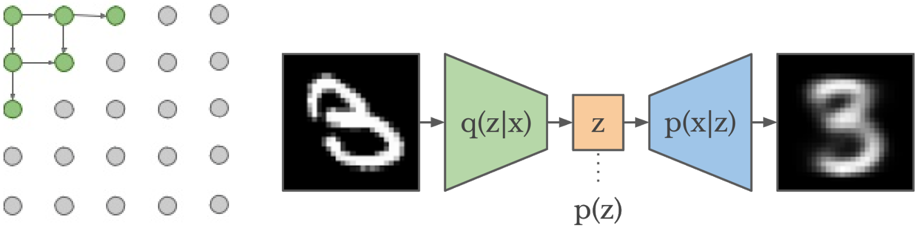


Figure 2.4: **DensePose** aims at mapping all human pixels of an RGB image to the 3D surface of the human body. *Left:* The image and the regressed correspondence in DensePose-RCNN. *Middle:* DensePose Coco annotations. *Right:* Partitioning of the body surface. Figure Source: Güler *et al.* [2]

as predicting a dense correspondence between 2D RGB images and a 3D surface based representation of the human body. They divide the human body into 25 parts, as seen in Figure 2.4, and with a regression head similar to the keypoint head in Mask-RCNN, they predict pixel correspondence to each part. Similar to OpenPose, they use an iterative refinement to improve performance. Further, they exploit the benefit of multi-task learning by utilizing information from related tasks, such as keypoint estimation and instance segmentation. This results in a model that can generate highly-accurate dense correspondences between images and the body surface in multiple frames per second. Their method operates at 4-5 FPS for 800 × 1100 resolution, but their runtime scales linearly on the number of persons in the image, just as Mask R-CNN.

2.2 Generative Models

Generative models is a subfield of unsupervised learning where the goal of the model is to represent a probability distribution over multiple variables in some way. Some generative models allow evaluating the probability distribution explicitly. Others do not allow the evaluation of the probability distribution, but it supports operations that implicitly requires knowledge about the distribution, such as drawing samples from it. In our task, we are interested in having a latent vector z and the conditional information y, then sample from the probability distribution to generate our new image x; we desire to model $p(x|z, y)$. There exist several methods to approximate this, and the most relevant techniques are Pixel-RNN [53], Variational Autoencoders [31], and Generative Adversarial Networks [14].



(a) **PixelRNN** starts in a corner of an image and generates each pixel sequentially.

(b) **VAE** consists of an encoder $p(z|x)$ and a decoder $q(x|z)$. x is the original image, and z is a latent variable that is optimized to follow a pre-defined probability distribution (e.g. gaussian)

Figure 2.5: Illustrations of how PixelRNN and Variational Autoencoders work.

PixelRNN explicitly defines the probability distribution, generating pixels in an image sequentially starting from the top left corner. With this definition, we can optimize the network with gradient descent by maximizing the likelihood of data; in practice, they perform softmax classification for each pixel into a class between [0-255]. Example of how PixelRNN generates pixels are shown in Figure 2.5a. The drawback of PixelRNN is its sequential nature, making training and inference time extremely slow. PixelCNN [53] improves training time by enabling parallel inference of pixels, but inference time is still sequential. Also, PixelRNN struggles to model complex data distributions, and the generated images are usually blurry.

Variational Autoencoder (VAE) is built on an encoder-decoder architecture and can be trained purely with gradient-based methods. Figure 2.5b illustrates the basic architecture of a VAE. VAE is an elegant, simple, and theoretical pleasing approach to minimize the Kullback-Leibler divergence (KL-divergence) between the original and the generated data distribution [13]. VAE explicitly defines a probability distribution; however, due to the intractability of the function, VAE are optimized on the variational lower bound. The main drawback of VAEs is its tendency to generate blurry images. The causes of this are yet not known , but this is an issue shared with generative models that optimize a log-likelihood, or the KL-divergence [13].

Generative Adversarial Networks define an implicit probability distribution and try to optimize the generative model as a two-player game [14]. We will discuss GANs in detail in Section 2.3.

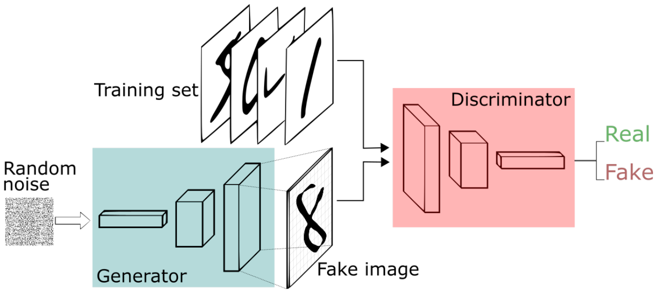


Figure 2.6: **Generative Adversarial Networks** consist of a generator that generates samples similar to the training set and a discriminator that tries to predict if the sample is real or fake. Figure source: https://skymind.ai/wiki/generative-adversarial-network-gan

2.3 Generative Adversarial Networks

Generative Adversarial Networks (GANs) [14] is a highly successful architecture to model a natural image distribution. GANs enables us to generate new images, often indistinguishable from the real data distribution. It has a broad diversity of application areas, from general image generation [29, 8, 80, 30], text-to-photo generation [81], style transfer [23, 58], and much more. With the numerous contributions since its conception, it has gone from a beautiful theoretical idea to a tool we can apply for practical use cases.

The basic idea of GANs is to set up a zero-sum game between two neural network adversaries. One of the players is called the generator, G, whose main objective is to create samples that should resemble the training data as closely as possible. The opposing player is the discriminator, D, whose job is to examine these samples and determine whether they are real (originate from the training data) or fakes (produced by the generator). Figure 2.6 illustrates a typical GAN setup.

Formally, the GAN objective function in its original form is given by:

$$\min_G \max_D \mathbb{E}_{x \sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))], \tag{2.1}$$

where $\boldsymbol{z} \in \mathbb{R}^d$ is a latent vector, with size d, drawn from a random noise distribution p_z (such as $p_z \in \mathcal{N}(0,1)$), and $\boldsymbol{x}$ is a sample drawn from our real data distribution p_{data}. G is the generator, and D is the discriminator, where the out-

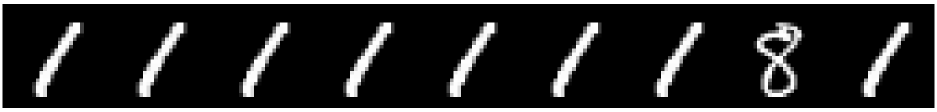


Figure 2.7: **Mode collapse on the MNIST dataset** [34]. The generator can generate a limited diversity of samples while still being able to fool the discriminator.

put of D is the predicted probability that the sample is real or fake, restricted by the sigmoid function. The objective function involves finding the Nash Equilibrium for this two-player min-max game. The min-max game will ideally force the generator to generate samples that look like perfect counterfeits, indistinguishable from the original data distribution and trick D to label them as real.

Without auxiliary information, this training procedure is remarkably brittle, requiring fine-tuned hyperparameters and careful architecture choices. Stable and efficient training for GANs is still an open research area; however, several improvements to objective functions, architecture choices, and the overall training process has improved GANs significantly.

Problems with original GANs

Original GANs suffered from a notoriously unstable training phase, often leading to unrealistic or low diversity images. One of the leading causes of this is *vanishing gradients* in the generator. Arjovsky *et al.* [5] proves that the generator suffers from vanishing gradients if the generated images are far from the original data distribution. This is due to the original objective function (Equation 2.1), which is shown to minimize the Jensen-Shannon divergence (JS-divergence) with an optimal discriminator [14]. However, the JS-divergence flattens out in cases when the generated data distribution is far apart from the real data distribution, leading to vanishing gradients in the generator.

The second cause of unstable training is something known as *mode collapse.* Mode collapse occurs when the generator can generate a limited diversity of samples while still being able to fool the discriminator. Figure 2.7 shows a simple example of mode collapse. The severity of mode collapse can vary from a complete collapse where the generator is independent of the noise variable and generates a single output sample, or a partial collapse where all the samples share similar features. Mode collapse is still a significant problem for modern GANs and an actively researched area.

2.3.1 GAN Objective Functions

The original objective function for GANs (Equation 2.1) is challenging to optimize without causing vanishing gradients or mode collapse. Since then, a large amount of objective functions has been proposed: Wasserstein GAN [5, 17], Relativistic GAN [27], Least Square GAN [44], and GAN with gradient penalties [57] are the most popular objective functions. In the following sections, we will take a closer look at two versions of Wasserstein GANs.

Wasserstein GANs

Arjovsky *et al.* [5] address several of the issues introduced by the original objective function, shown in Equation 2.1. They propose a new objective function based on Earth Mover's Distance (EM-Distance), indicating that EM-Distance can converge in cases when JS-divergence cannot. They propose a GAN based on minimizing this EM-Distance , called a Wasserstein GAN. The new Wasserstein objective function is given by

$$\max_{w \in W} \mathbb{E}_{x \sim p_{data}}[f_w(x)] - \mathbb{E}_{z \sim p(z)}[f_w(G(z)] \tag{2.2}$$

where $\boldsymbol{z} \in \mathbb{R}^d$ is a latent variable drawn from a random noise distribution p_z, and $\boldsymbol{x}$ is drawn from our real data distribution p_{data}. $\boldsymbol{G}$ and $\boldsymbol{f_w}$ is the generator and the discriminator, respectively. In their paper they refer to f_w as the critic; however, we will stick to the discriminator for simplicity. The discriminator is not restricted by a sigmoid function here ($f_w \in \mathbb{R}$), but it is required that $\{f_w\}_{w \in W}$ are all K-Lipschitz functions for some K. To roughly approximate the K-Lipschitz criteria, they ensure that the weights of the discriminator lie in a compact space after each gradient update, also known as weight clipping.

The Wasserstein GAN solves several challenges with training GANs, especially the problem of vanishing gradients in the generator. In cases where the generated images are far from the real data distribution, the gradient updates are still reliable. This enables us (and is encouraged) to train the discriminator till optimality [5]. The more we train the discriminator, the more reliable the gradients of the Wasserstein we get. Furthermore, they argue that an optimal discriminator makes it impossible for mode collapse. In practice, this doesn't seem to be the case. Several papers struggle with a lack of diversity (partial mode collapse) in their generated samples, leading to additional "tricks" to encourage sample diversity [29].

Improved Wasserstein GAN

Arjovsky *et al.* [5] states that weight clipping is a terrible way to enforce the K-Lipschitz constraint and encourage further work in this area. Gulrajani *et al.* [17] proposes an alternative to clipping weights: penalize the norm of the gradient of the discriminator with respects to its output. They illustrate that weight clipping can lead to undesired behavior on toy datasets, and replacing it with gradient penalty achieves state-of-the-art results on CIFAR-10 [32] and LSUN [78] (March 2017). The improved Wasserstein objective function is given by

$$L = \underbrace{E_{z\sim p(z)}[D(p_g)] - \mathbb{E}_{x\sim p_{data}}[D(x)]}_{\text{Original discriminator loss}} + \underbrace{\lambda\,\mathbb{E}_{\hat{x}\sim p_{\hat{x}}}[(||\nabla_{\hat{x}}D(\hat{x})||_2 - 1)^2]}_{\text{Gradient penalty}} \tag{2.3}$$

where $\boldsymbol{z} \in \mathbb{R}^d$ is a latent variable drawn from a random noise distribution p_z, and $\boldsymbol{x}$ is drawn from our real data distribution p_{data}. $\boldsymbol{G}$ and $\boldsymbol{D}$ is the generator and the discriminator, respectively. They define a sampling $p_{\hat{x}}$ distribution that samples evenly along straight lines between a pair of points from the data distribution p_{data} and p_g. Their proposition gives the thought behind this sampling; the gradient norm is 1 almost everywhere along the straight lines between p_{data} and p_g. Enforcing the constraint along these lines is sufficient to enforce the K-Lipschitz constraint on the discriminator, and the presented results are a significant improvement.

2.3.2 Conditional GANs

In a generative adversarial network, there is no control over modes of the data being generated; for example, it is impossible to tell our generator to generate an image with a cat. Mirza *et al.* [46] proposes a novel approach to generate samples based on auxiliary information. Instead of generating images from a random noise variable z, we combine this latent vector with additional conditional information.

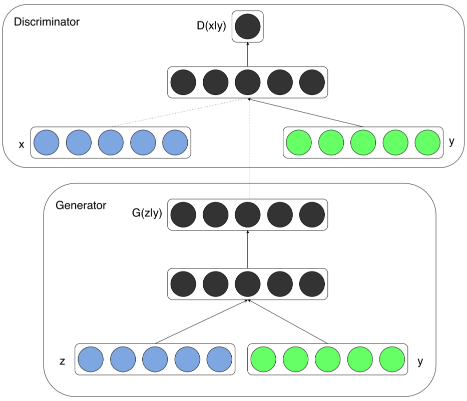


Figure 2.8: **Conditional Generative Adversarial Network**. x is the original sample, y is the conditional information and z is the latent vector. Figure source: [46].

This information could be anything such as class labels, a base image to perform image inpainting, or even data from other modalities to perform tasks such as segmentation. Figure 2.8 shows a conditional GAN architecture.

In practice, converting a GAN to a conditional GAN is fairly simple. For the method proposed by Mirza *et al.* [46], the only modification required is to provide the conditional information to both the generator and discriminator. Other approaches include Auxiliary Classifier GANs (AC-Gans) [52], where the conditional information is provided to the generator; however, the information is not provided to the discriminator explicitly. The discriminator is provided the information implicitly, by introducing an additional output head and objective function to train the discriminator to predict the conditional information. Odena *et al.* [52] shows that AC-Gans is a well-suited approach for datasets with a large variety of conditional image class information.

2.3.3 Curriculum Learning with GANs

Curriculum learning of GANs is based on the idea that learning to complete a task is easier to learn in steps. You can think about it as if you want to learn how to multiply numbers, you first have to learn what addition is. Curriculum learning of GANs is to learn the complex mapping from latent vectors to high-resolution images in steps. First, you want your model to learn high-level

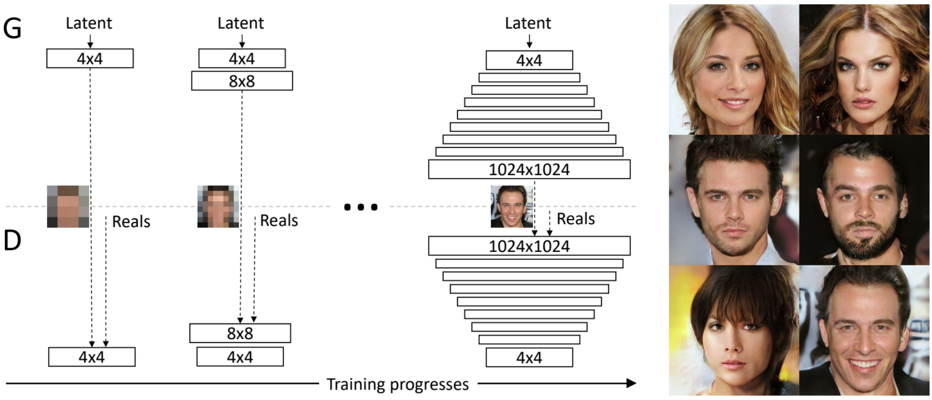


Figure 2.9: **Progressive growing of GANs**. The training starts with a low spatial resolution of 4x4, then they incrementally add new layers to both the generator (G) and discriminator (D), thus increasing the spatial resolution of the generated images. On the right is six examples generated by their model at 1024x1024. Figure source: Karras *et al.* [29]

structures of the image distribution, such as shape and general color. Later on, you want the model to shift its attention to finer scale details. Several authors explore this basic idea in different ways; Wang *et al.* [71] explores using multiple discriminators at different spatial resolutions, StackGANs [81] defines a generator and discriminator for each level in an image pyramid, and progressive growing of GANs [29] iteratively increases the spatial resolution. The latter method has shown impressive results on generating close to photo-realistic images of faces and images based on the CIFAR-10 [32] dataset.

Progressive Growing GANs

Progressive growing of GANs [29] is a novel training method that improves training stability, image quality, and covergence time. The technique builds upon the observation that a complex mapping from latent to high-resolution images is easier to learn in steps. The method initially starts with low-resolution images (4x4), then progressively increase the resolution by adding new layers to both the discriminator and the generator. This incremental approach enables the training procedure first to discover the large-scale structure of the image distribution, then shift its attention to increasingly finer scale detail. Figure 2.9 shows the overall training architecture.

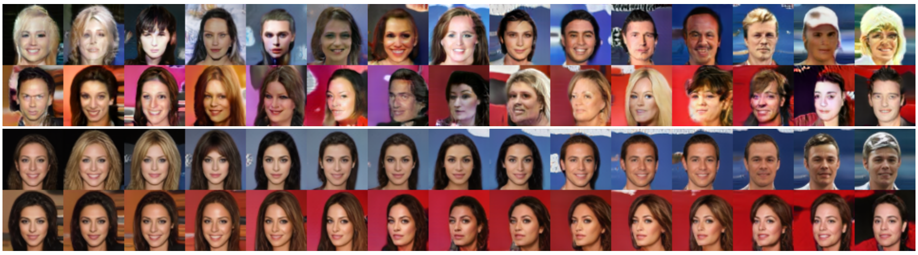


Figure 2.10: **The effect Exponential Moving Average (EMA) for GANs.** Generation for the CelebA dataset for the same two noise samples from 50k to 200k with 10k intervals. Top two rows are generated images without EMA; bottom two rows with EMA ($\beta = 0.9999$). Figure source: Yazici *et al.* [76]

Progressive training has several benefits. Early on, the generation of smaller images is significantly more stable because there are less class information and fewer modes. By increasing the resolution incrementally, we are continuously asking much simpler questions compared to the end goal of discovering a mapping from latent vectors to high-resolution images. An additional benefit of this approach is reduced training time. With a progressive training approach, we perform the majority of gradient steps at a lower resolution. The author states that images of comparable quality are obtained up to 2-6 times quicker, compared to training solely on the target resolution.

2.3.4 The Unusual Effectiveness of Averaging in GANs

Exponential Moving Average (EMA) of the generator is a technique that has shown to be successful in removing artifacts and improving image quality for GANs. EMA has a copy of the generator parameters that are updated after each training iteration with the following equation:

$$\theta_{EMA}^{(t)} = \beta\theta_{EMA}^{(t-1)} + (1-\beta)\theta^{(t)}, \tag{2.4}$$

where $\theta_{EMA}^{(t)}$ is the parameters of the moving average of the generator at time step t, and $\theta^{(t)}$ is the parameters of the generator at time step t. β is a scalar we have to set between 0 and 1 (often $\beta = 0.999$). Note that the moving average of the generator is newer used for a training step, and is only updated by Equation (2.4).

EMA has seen a wide adoption recently [29, 30, 8, 40], and it generally improves image quality. Yazici *et al.* [76] does a large scale study of the technique and presents results indicating that EMA improves image quality on various datasets, network architectures, and GAN objectives. Figure 2.10 shows the effect of EMA on the CelebA dataset. Notice, the top row has several artifacts and a substantially worse image quality compared to the bottom row with EMA .

2.4 Evaluation Metrics for Generative Models

Overall image quality is a subjective measurement, but having a quantitative metric to evaluate a generative model is a vital requirement to drive algorithm research. Evaluating synthetic images from a generative model is a difficult challenge, and currently, we have no optimal measurement for image quality. To assess a generative model, we desire two properties: First, it should generate high-quality images that are indistinguishable from our original dataset. Secondly, the generative model should be able to sample from a large part of the training distribution; in other words, there is a broad diversity in the generated images. In recent years, a variety of metrics have been used to evaluate image quality and diversity; Multi-Scale Structural Similarity [72], Multi-Scale Statistical Similarity [29], Inception Score (IS) [60], and Frèchet Inception Distance (FID) [21]. In the following sections, we will take a closer look at IS and FID. Also, we will review the recent advances in the area; precision and recall for generative models.

2.4.1 Inception Score

Inception Score (IS) [60] is a method to automatically evaluate both the image quality and the sample diversity of generated images. The metric has become a popular evaluation technique due to its property of correlating well with human evaluation. For an image x and a predicted image class y, IS evaluates two properties:

1. Images that contain meaningful objects should have a conditional label distribution $p(y|\boldsymbol{x})$ with low entropy. In other words, the inception network should be highly confident that there is a single object in the image.

2. The generative model should output a high diversity of images from all the different classes in ImageNet: $p(y)$ should be high for all classes y.

IS applies a pre-trained Inception Network v3 [66] to every generated image to get the conditional label distribution $p(y|\boldsymbol{x})$, and calculates the statistics of the network's output. Formally the equation of the IS is given by:

$$\boldsymbol{IS}(G) = exp(\mathbb{E}_{x\sim p_g}\, KL(p(y|\boldsymbol{x}) \,||\, p(y))), \tag{2.5}$$

where KL is the KL-divergence between two distributions, and $p(y) = \int_x p(y|\boldsymbol{x})p_g(x)$ is the marginal class distribution. x is the generated image and y is the predicted class probabilities of the inception network. In practice, a high IS indicates that the generated images clearly contain a single object, and the images contain a large variety of different object classes. A small IS indicates that the generated images contains very few unique object classes, and the images do not clearly include a single object class.

IS is widely used; however, it has several issues:

1. **Intra-class mode collapse:** IS is insensitive to intra-class mode collapse. If the generative model suffers from a mode collapse for each class, meaning that it only generates one image per class, it would give $p(y)$ a uniform distribution, and $p(y|\boldsymbol{x})$ would not be affected, giving us a high inception score.

2. **Sensitivity to weight:** IS depends on the parameters of the Inception network. Salimans *et al.* [60] show that the IS mean can vary with as much as 11% depending on the deep learning framework (Tensorflow vs. Keras vs. Torch).

3. **Usage beyond ImageNet:** The inception network is pre-trained on the ImageNet dataset; however, it is frequently applied to other datasets. The usage beyond ImageNet can give misleading IS.

4. **Overfitting of generator:** A generative model that memorizes a subset of the training data, would perform exceptionally well in terms of IS [6].

5. **Score calculation:** The final score calculation is dependent on the number of batches the generated images are split into [6]. IS is an average of the KL-divergence over a sequence of batches, which is an approximation of the exact IS.

6. **IS does not look at the real data distribution.** IS is based on statistics calculated from the generated images, giving us no information if the images are realistic in comparison to the original training images.

2.4.2 Frèchet Inception Distance

Frèchet Inception Distance (FID) [21] is an alternative approach to IS, solving some of the issues with the previous metric. FID includes the original data distribution in its evaluation, it can detect intra-class mode dropping, it is shown to be more robust to noise than IS, and results show a negative correlation between FID and visual quality of the generated images.

To quantify the quality of generated images, FID first embeds images into a feature space given by a layer of the Inception Net. Then, viewing the embedding as a multivariate Gaussian, the mean and covariance are estimated for both the generated data and the real data. Frèchet Distance is then calculated between these two Gaussians. Formally, FID is given by:

$$\boldsymbol{FID}(x,g) = ||\mu_x - \mu_g|| + Tr(\Sigma_x + \Sigma_g - 2(\Sigma_x\Sigma_g)^{\frac{1}{2}}), \tag{2.6}$$

where (μ_x, Σ_x) and (μ_g, Σ_g) are the mean and covariance of the sample embeddings from the data distribution and model distribution, respectfully. Tr is the sum of the diagonal. Lucic *et al.* [43] provides a thorough empirical analysis of the FID metric, and they show results arguing for FIDs ability to detect intra-class mode dropping.

FID has proven to be an essential metric to evaluate image quality and diversity in images. However, it still suffers from several of the disadvantages present in the IS. The FID is sensitive to weights, it assumes an Inception Network pre-trained on ImageNet, and it is unable to detect overfitting of the generator.

2.4.3 Precision and Recall for Image Quality

IS and FID group two separate goals (image quality and diversity) in a single score without a clear trade-off. For example, a low FID may indicate realistic images with a large amount of variation, or anything between. Recently, Sajjadi *et al.* [59] proposed a metric that expresses the quality of generated samples using two separate metrics: *precision and recall.* Precision corresponds to the average sample quality, while recall corresponds to the coverage of the training distribution. This separation gives us the ability to evaluate the generated images and selectively choose a model with larger diversity or better image quality.

Kynkäänniemi *et al.* [33] proposes an alternative formulation to measure preci-

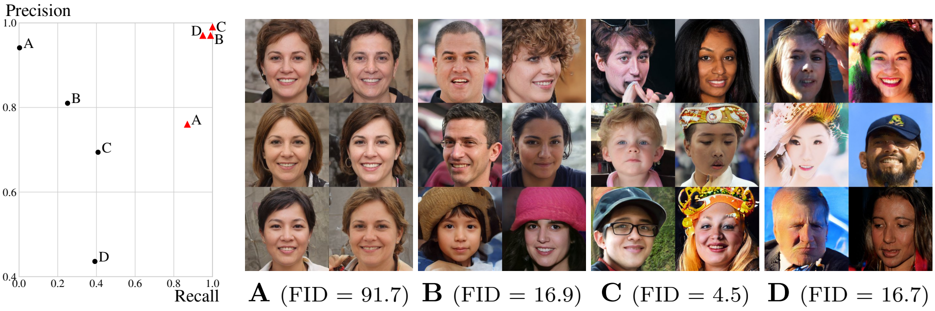


Figure 2.11: **Comparison of precision-recall metric and FID** for different StyleGAN [30] setups (lower FID is better). Black dots show Kynkäänniemi *et al.* [33] precision and recall, and red triangles denote Sajjadi *et al.* 's [59] method. We recommend zooming in to better assess the quality of images. Figure source: Kynkäänniemi *et al.* [33].

sion and recall for generated images. They present their results on state-of-the-art GANs for ImageNet [8] and human face generation [30]. Figure 2.11 shows a precision-recall curve along with FID values. The precision-recall curve shows clear correlation between precision and image quality, and recall and image diversity. Also, note that Sajjadi *et al.'s* method tends to give over-optimistic results, and it cannot correctly interpret situations where a large number of generated samples are packed together.

2.5 Mixed Precision Training

To utilize tensor cores on NVIDIA GPUs, we need to satisfy a set of requirements: the batch size and convolutional kernel have to be divisible by 8, and the floating point operation has to be done in 8 or 16 bit. The first requirement is rather simple to achieve. However, training a neural network with floating point 16 precision (FP16) or lower, the final results are often degraded in comparison to FP32. Micikevicius *et al.* [45] proposes a method called *mixed precision training*, where they try to do all possible computations in FP16 where it is safe and FP32 otherwise. Micikevicius *et al.* presents a range of experiments on a diversity of classification and object detection datasets with state-of-the-art models. With mixed precision training, they report comparable results to fp32 training with $2 - 6x$ speedup with a significant reduction in GPU memory usage.

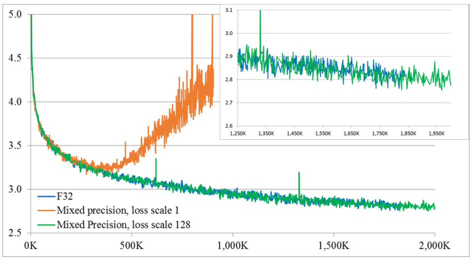


Figure 2.12: The effect of static loss scaling for mixed precision training. Scaling the loss with a static loss scale reduces overflow in gradient computations, making the loss converge. The loss graph is an LSTM trained for language modeling. Figure source: Micikevicius *et al.* [45]

Miciekvicius *et al.* introduces three techniques to make mixed precision training possible: maintaining a master copy of weights in fp32, loss-scaling that prevents underflow in gradients (gradients becoming 0), and fp16 arithmetic with accumulation in fp32. Figure 2.12 shows the effect of a static loss scale for training an LSTM for language modeling. Since then, empirical experiments have shown other useful techniques to improve mixed precision training. For example, the APEX framework [51] recommends performing batch normalization and other precision-sensitive computations in fp32, and they recommend a dynamic loss scale instead of a static scale.

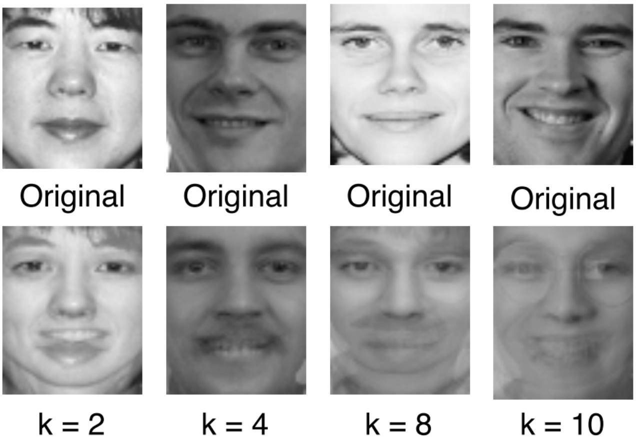


Figure 2.13: **K-Same face de-identification**. K-same family of algorithms finds the k most similar images, then anonymize the image by taking the average of the k-same images. Figure Source: Gross *et al.* [15]

2.6 Related Work

There exists a limited number of research studies on the task of removing privacy-sensitive information from an image. Typically, the approach chosen is to alter the original image such that we remove all the privacy-sensitive information. These algorithms can be applied to all images; however, we have no assurance that we remove the privacy-sensitive information. Naive methods that use simple image distortion have been discussed numerous times in literature [7, 49, 15, 50, 16], such as pixelation and blurring. Simple image distortion methods are inadequate for removing the privacy-sensitive information [16, 49, 50], and they alter the data distribution substantially.

K-Same Family of Algorithms

The K-same family of algorithms [50, 28, 16] implements the k-anonymity algorithm [65] for face images. The basic idea behind the K-same methods is to find similar faces to the original face, then anonymize the original face by replacing it with the average of the similar images. Figure 2.13 shows faces anonymized by a K-same method. Newton *et al.* [50] proves that the k-same algorithm can remove all privacy-sensitive information; but, the resulting images often contain

”ghosting” artifacts due to small alignment errors [16].

Jourabloo *et al.* [28] extends the k-same algorithm for face de-identification of grayscale images to preserve a set of facial attributes, such as ethnicity, pose, age, and gender. This is different from our work, as we do not directly train our generative model to generate faces with similar attributes to the original image. In contrast, our model performs complex semantic reasoning to generate a face that is coherent with the overall context information given to the network, yielding a realistic face given the context.

GANs for Face Anonymization

Ren *et al.* [55] proposes a method that anonymizes faces in videos. They use a GAN that tries to alter a face to remove all privacy-sensitive information. Their results are impressive in both the image and video domain; however, they base the generated face on the original face, and they do not discuss if their method removes all privacy-sensitive information.

Further, the transition between generated parts and original parts of the images are visible in several of their examples, which is undesirable. In contrast to their method, we can ensure the removal of all privacy-sensitive information, as our generative model never observes the original face.

Image Inpainting

Image Inpainting is a closely related task to what we are trying to solve, and it is a widely researched area for generative models [39, 26, 36, 77]. Several research studies have looked at the task of face completion with a generative adversarial network [36, 77]. They mask a specific part of the face and try to complete this part with the conditional information given. Figure 2.14 shows two examples of this. From our knowledge, and the qualitative experiments they present in their papers, they are not able to mask a large enough part to remove all privacy-sensitive information. As the masked region grows, it requires a more advanced generative model that understands complex semantic reasoning, making the task considerably harder. In comparison to these methods, we expect a rectangular shaped mask to identify the area we want to inpaint. Also, we use a more complex dataset than the typical CelebA dataset [42].

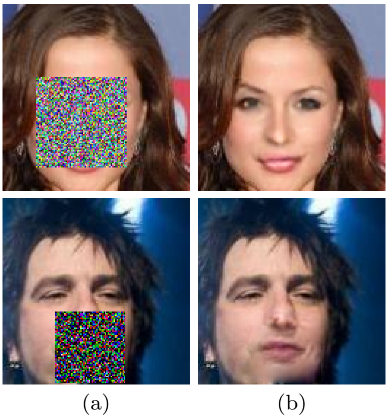


Figure 2.14: **Image inpainting of faces**. In each row from left to right: (a) masked input. (b) face completeion result. Figure source: Li *et al.* [36]

DeepFakes

The subject of DeepFakes has gotten a lot of attention in the last years in the media. The Face2Face [67] method was a groundbreaking result in this area, presenting a model that was able to portray a person speaking. Given a source video of a person speaking and a target video of the person we desire to portray, they were able to re-enact the person's lip movement photo-realistically. Further work on re-enacting lip movement to sync up to audio has given us videos that are unidentifiable to be a generated or fake video sequence. Suwajanakorn *et al.* [64] simplifies the task, by generating the lip movement directly from the audio and applying this to the target video. The result of this is truly remarkable and indistinguishable from the original video [1]. These results have inspired communities to develop algorithms to swap out a persons face in a video.

Face-swap [1] is a community-driven method for swapping faces between videos, and it has attracted a lot of attention the recent year for their impressive results. They can generate a video with swapped faces with a seamless transition between the generated parts of the image and the original image[2]. The original model is a shallow autoencoder optimized on the L_1 loss and further models have been implemented based on GANs and VAE. The base input is a sequence of images from the source face, the target image, and the facial pose of the target image.

[1] Synthesizing audio video example: https://youtu.be/9Yq67CjDqvw
[2] Face-swap example video: https://youtu.be/r1jng79a5xc

The downside of using the face-swap method is its dependency on robust annotations of the pose of the face. Also, the method base the generation on the original target image, giving us no assurance that the resulting face is anonymized or not.

Chapter 3

Method

In this chapter, we introduce our architecture and discuss the design choices behind them. Then, we will introduce a new dataset, the FDF Dataset.

3.1 Architecture Overview

Our proposed architecture, *DeepPrivacy*, consists of two separate stages. The first stage is responsible to detect all privacy-sensitive regions in the image and estimate the pose of the individual. The second stage is a generative model responsible for generating a new anonymized face. To handle irregular poses, we use the pose of the face as conditional information to guide the model.

3.1.1 Face Detection and Pose Estimation

Face detection and pose estimation is the cornerstone of our proposed architecture. For face detection, we require a model that is able to detect all privacy-sensitive faces (has high recall), and for pose estimation, we need an exact pose annotation. The focus of this thesis is on developing a generative model; therefore, we reviewed the current models for face detection and pose estimation that are available open-source. We chose models with simple-to-use implementation and state-of-the-art performance on the selected tasks. Further development or

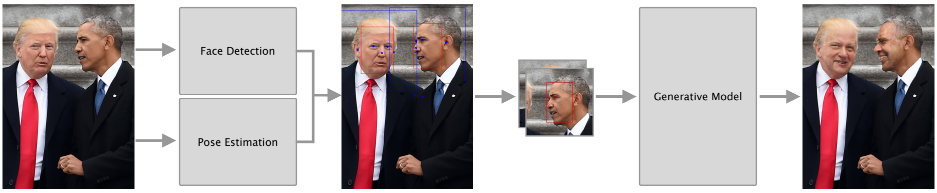


Figure 3.1: **Overview of the DeepPrivacy architecture**. Initially, the face is detected with S3FD [82] and pose estimation is done with Mask R-CNN [19]. The bounding box and pose of the face are given as conditional information to the generator, which generates an anonymized face for each person.

training of these models are outside of the scope for this thesis.

Face Detection is done with DSFD [35]. DSFD achieves state-of-the-art results on the WIDER-Face dataset. The model uses a ResNet-152 backend, and achieves 90.4% AP on the WIDER-Face dataset. For inference on a single image, we predict bounding boxes for the original image, a horizontally flipped image, and for two different image scales (50% and 150% of original size). We apply non-maxiumum supression to these predictions, and we filter with a bounding box confidence threshold of 0.3.

The DSFD authors claim that their model can achieve 22 FPS with a ResNet-50 backbone on an NVIDIA P40 GPU during inference. However, their officially released model is a ResNet-152 and they report no inference time for this model. In practice, we experience about 3-4 FPS (including all flipping/scaling predictions) with an NVIDIA V100-32GB. These numbers should be taken with a grain of salt, as we do no pipeline optimization.

Pose estimation is done with a Mask R-CNN [19] using three output branches; segmentation, object detection, and keypoint estimation. The Detectron repository [12] contains several Mask R-CNN models with different feature extractors, where all models are trained on the COCO dataset. We chose the feature extractor with the highest AP for keypoint estimation. This model has a ResNeXt [74] backbone with 101 layers utilizing feature pyramid networks [37]. This model achieves 67.0% AP for pose estimation on the Coco minival 2014 dataset. The inference time is approximately $0.394s$ per image.

Matching of keypoints and face detections

To match each keypoint with a corresponding bounding box we use a greedy approach. We assume that each keypoint has a single unique bounding box match and vice versa. We sort both bounding boxes and keypoints based on descending prediction confidence, then iterate through them sequentially. For each potential match, we check if the nose and eye keypoint is within the bounding box of the face; if this is the case, we register it as a match. We disregard any bounding box or keypoint without a match. Algorithm 1 shows the pseudocode for our matching strategy.

Algorithm 1 Algorithm to match keypoints and bounding box predictions.

```
procedure MatchAnnotations(keypoints, bounding_boxes)
    sort bounding_boxes descending on prediction confidence
    sort keypoints descending on prediction confidence
    matches ← []                    ▷ Initialize matches to an empty list
    for bbox in bounding_boxes do
        for kp in keypoints do
            if kp in bbox then
                matches add match(bbox, kp)
                mark bbox as matched
                mark kp as matched
    return matches
```

3.2 Generative Model

Our proposed generative model is a conditional GAN, generating images based on the surroundings of the face and sparse pose information. We ground our model on the model proposed by Karras *et al.* [29]. Their model is a non-conditional GAN and we perform several alterations to include conditional information. Figure 3.2 shows the overall architecture of our GAN. We use the surrounding background and pose of the face as conditional information, which is given to both the discriminator and generator.

We use seven facial landmarks to describe the pose of the face, including the following keypoints: left/right eye, left/right ear, left/right shoulder, and nose. To reduce the number of parameters in the network, we pre-process the pose information into a one-hot encoded image of size $K \times M \times M$, where K is the

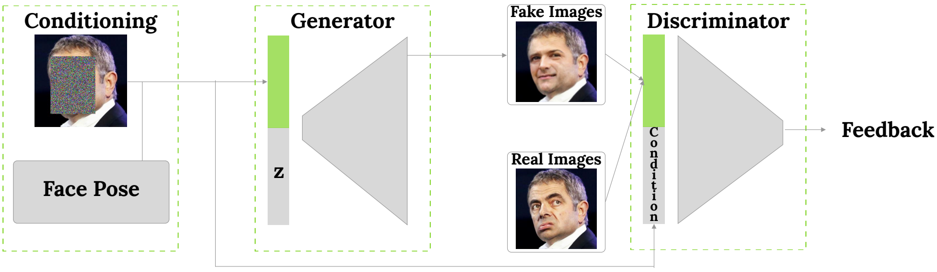


Figure 3.2: **Our GAN architecture.** We use a conditional generative model, where information about the surrounding and pose of the face is given to both the generator and discriminator.

number of landmarks and M is the target resolution.

Progressive growing training technique improves the final image quality and overall training time, and it's crucial for our model's success. We apply progressive growing to both the generator and discriminator to grow the networks from a starting resolution of 8. We double the resolution each time we expand our network until we reach the final resolution of 128 × 128. To include the pose information through the whole training period, we decided to include this information for each resolution for both the discriminator and generator.

3.2.1 Generator Architecture

Figure 3.3 shows our proposed generator architecture for 128 × 128 resolution. Our generator uses U-net [56] architecture to include background information, similar to the generator proposed by Isola *et al.* [24]. The encoder and decoder have the same number of filters in each convolution, but the decoder has an additional 1×1 bottleneck convolution after each skip connection. This bottleneck design reduces the number of parameters in the decoder significantly. To include the pose information for each resolution, we concatenate the output after each upsampling layer with pose information and the corresponding skip connection. The general layer structure is identical to Karras *et al.* [29], where we use pixel replication for upsampling, pixel normalization, and LeakyReLU after each convolution, and equalized learning rate instead of careful weight initialization. In total, the generator has $46.9M$ parameters.

Progressive growing: Each time we increase the resolution of the generator,

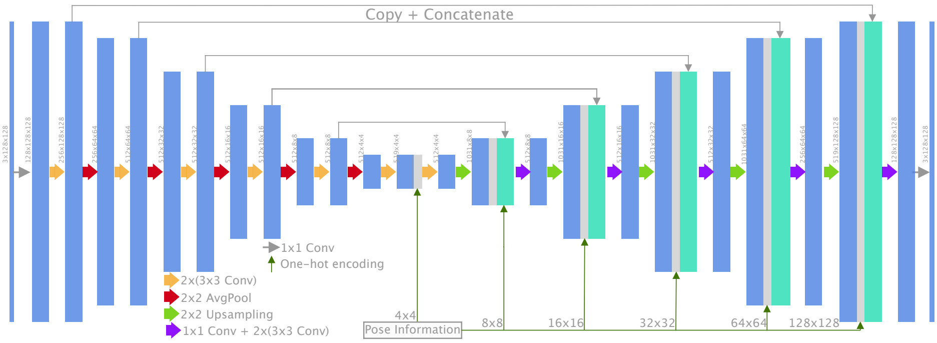


Figure 3.3: **Our generator architecture** for 128x128 resolution. Each convolutional layer is followed by pixel normalization [29] and LeakyReLU($\alpha = 0.2$). After each upsampling layer, we concatenate the upsampled output with pose information and the corresponding skip connection.

we add two 3 × 3 convolutions to the start of the encoder and to the end of the decoder. We use a transition phase identical to Karras *et al.* [29] for both of these new blocks, making the network stable throughout the training. We note that the network is still unstable during the transition phase, but it is significantly better compared to training without progressive growing.

3.2.2 Discriminator Architecture

Our proposed discriminator architecture is identical to the one proposed by Karras *et al.* [29], with a few exceptions. First, we include the background information as conditional input to the start of the discriminator, making the input image have six channels instead of three (2xRGB). Secondly, we include pose information at each resolution of the discriminator. The pose information is concatenated with the output of each downsampling layer, similarly to the decoder in the generator. Finally, we remove the mini-batch standard deviation layer presented by Karras *et al.* [29], as we find the diversity of our generated faces satisfactory. Otherwise, our discriminator uses the same structure in all layers. Table 6.1 in Appendix A includes a detailed description of the discriminator architecture.

The adjustments made to the generator doubles the number of total parameters in the network. To follow the design lines of Karras *et al.* [29], we desire that the complexity in terms of the number of parameters to be similar for the discriminator and generator. We evaluate two different discriminator models, which we will name the *deep discriminator* and the *wide discriminator*. The deep discriminator doubles the number of convolutional layers for each resolution. To mimic the skip-connections in the generator, we wrap the convolutions for each resolution in residual blocks. The wider discriminator keeps the same architecture; however, we increase the number of filters in each convolutional layer by a factor of $\sqrt{2}$. For example, for a convolutional block with initially 512 filters, we use 724 filters instead. In total, the wide discriminator has $45.6M$ parameters, and the deep discriminator has $44.7M$ parameters.

3.3 Mixed Precision Training for GANs

Training a generative adversarial network with mixed precision without degrading the final results is complicated. GANs include several different loss functions, all with varying loss scales. The variety of loss scales makes it almost impossible to have a single loss scaling factor for mixed precision without impacting the convergence of our model. Therefore, we decide to use a unique loss scaling factor for each loss. Also, the gradient penalty term in the Wasserstein loss (Equation 2.3) includes the computation of second derivatives, which is hard to perform in fp16. Therefore, we scale the logits of the discriminator before computing the first derivative. From this, we can calculate the gradients (and the second derivative of the gradient penalty term) without impacting convergence.

For our GAN, we utilize mixed precision training with Pytorch and the Apex framework [51]. From experiments, we observe no difference in convergence or final results, and we observe a 220% speedup in comparison to pure fp32 training. Also, we notice a significant reduction in GPU memory usage, allowing us to increase the batch size by approximately 50%.

3.4 Flickr Diverse Faces Dataset

FDF (Flickr Diverse Faces) is a new dataset of human faces, crawled from the YFCC-100M dataset [68]. It consists of 1.3M human faces with a minimum resolution of $128x128$, containing facial landmarks and a bounding box annotation for each face. The dataset has a vast diversity in terms of age, ethnicity, facial pose, image background, and face occlusion. The dataset was mainly extracted from scenes related to traffic, sports events, and outside activities. In comparison to the FFHQ dataset [30], our dataset is largely more diverse in facial poses, and

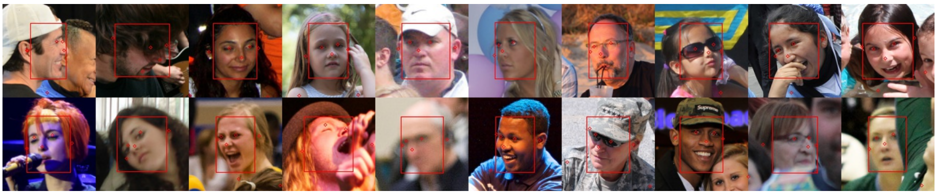


Figure 3.4: The FDF dataset. Each image has a sparse keypoint annotation (7 keypoints) of the face and a tight bounding box annotation. We recommend the reader to zoom in.

it is generally much larger; however, the FFHQ dataset has a higher resolution. Figure 3.4 shows a randomly picked set of samples from the FDF dataset with keypoint and bounding box annotation.

The FDF dataset is a high-quality dataset with few annotation errors. The faces are automatically labeled with state-of-the-art keypoint and bounding box models, and we use a high confidence threshold for both the keypoint and bounding box predictions. The faces are extracted from $964,099$ images in the YFCC100-M dataset. For keypoint estimation, we use Mask R-CNN [19], with a ResNet-50 FPN backbone [37]. For bounding box annotation, we use the Single Shot Scale-invariant Face Detector (S3FD) [82]. Keypoints and bounding boxes are thresholded and matched together with the same approach as explained in Section 3.1.1. Due to the large number of images we want to annotate, we used different face detection and pose estimation models from what presented in Section 3.1.1 to improve inference time. Mask R-CNN with ResNet-50 (in comparison to a ResNeXt-101 backbone) improves GPU inference time by approximately 330% (322ms vs. 97ms) while maintaining a high AP on keypoint prediction (64.2% vs. 66.8% AP). S3FD can perform inference at about 36 images per second according to the authors; however, we observe about 5-7 FPS. S3FD degrades results somewhat in comparison to DSFD (84% vs. 90% AP), but the model is sufficient for our purpose.

Chapter 4

Experiments and Results

In this section, we will present our experiments and results. First, we will describe our experimental plan and experimental setup. Then, we will present our results analyzing the impact of anonymization on the WIDER-Face dataset. Finally, we present ablation experiments on the FDF dataset.

4.1 Experimental Plan

The goal of this thesis is to anonymize faces in images by replacing the face with a realistic face fitting the given situation. As previously discussed, evaluating a generative model with quantitative metrics is extremely hard, and it is a subjective task. Therefore, to evaluate our proposed model, we will perform several qualitative and quantitative experiments. First, we will perform a qualitative evaluation of the proposed model to assess the generated faces and how they fit the given conditional information. For quantitative assessment, we will anonymize the WIDER-Face validation set and evaluate the performance of a face detector on the anonymized dataset. This experiment will indicate how our proposed model performs on a large scale and how well our proposed model retains the original data distribution. Finally, we perform a variety of ablation experiments to discuss the architecture choices behind our model. The ablation experiments will be evaluated quantitatively with FID. All the presented results will be done on the validation set of FDF (consisting of 50K images), or other data not used for training. By using no training data for evaluation, we can ensure that our model generalizes to unseen images.

4.2 Experimental Setup

In this section, we will describe details about our data pre-processing pipeline, and the hyperparameters used for our final model. For further details, please revise the source code attached [1] with this thesis.

4.2.1 Dataset

FDF dataset is the foundation of our experiments. The FDF dataset is used for training and validating our generative model. The dataset is described in detail in Section 3.4. We split the dataset into a validation set of 50K images and a training set of 1.283M images. The FDF dataset will be publicly released with our BMVC paper. If required, contact me (hakon.hukkelas@ntnu.no) or my supervisor Frank Lindseth (frankl@ntnu.no) to get early access to the dataset [2].

We use the WIDER-Face [75] validation dataset to evaluate our model's ability to retain the original data distribution. The dataset has a high degree of variability in scale, pose, and occlusion. The WIDER-Face validation set is split into three different challenges: easy, medium, and hard based on the detection rate of EdgeBox [83]. The validation set consists of 3,220 images, with an average of 1.22 faces per image. Note, our generative model is never trained on any samples in the WIDER-Face dataset.

4.2.2 Training Details

The initial hyperparameters are based on results from Karras *et al.* [29], and we have used minimal time for hyperparameter tuning. We use the following batch size for the given resolutions: $8 \times 8 : 256$, $16 \times 16 : 256$, $32 \times 32 : 128$, $64 \times 64 : 72$, and $128 \times 128 : 48$. We use Adam Optimizer with a learning rate of 0.00175 and $\beta_1 = 0, \beta_2 = 0.999$. For each expansion of our network, we use a transition and stabilization face of $1.2M$ images each. We use EMA for the weights of the

[1] The code with this thesis can be downloaded from: Google Drive

[2] The FDF dataset will be released on Github: github.com/hukkelas/FDF

generator with a decay of 0.999. Our final model is trained for 18 days on two NVIDIA V100-32GB GPUs.

To expand our training dataset, we use simple data-augmentation techniques. All images are scaled to the range of (-1,1), and for the training dataset we use two random data augmentation methods; horizontal flip of the images, and 2% random shift of the face bounding box width and height.

Tensor Core Modifications

To utilize tensor cores in NVIDIA's new Volta architecture, we do several modifications to our network, following the requirements of tensor cores. First, we ensure that each convolutional block uses a number of filters that are divisible by 8. Secondly, we make certain that the batch size for each GPU is divisible by 8. Further, we use automatic mixed precision for Pytorch [51] to significantly improve our training time. We see an improvement of 220% in terms of training speed with mixed precision training.

4.2.3 Evaluation Details

For evaluations done on the FDF dataset, we use a validation set of 50K images for final assessment. FID is calculated using the official FID implementation by Heusel *et al.* [21]. The FDF dataset consists of images containing a single face, but several of these faces can come from the same original image. Therefore, we ensure that the validation set of FDF does not include faces that come from the same image as a face in the training set.

For the WIDER-Face dataset, we use the validation set for all three challenges: easy, medium and, hard. As the WIDER-Face dataset has bounding box annotations for each face, we do not need to use the face detection stage in our pipeline. The WIDER-Face dataset has no facial landmark annotation; therefore, we use Mask R-CNN to predict the keypoints. To match the landmarks with the annotated bounding boxes, we use the same greedy approach as described in Section 3.1.1. Mask R-CNN is not able to detect keypoints for all faces, especially in cases with high occlusion, low resolution, or faces turned away from the camera. Thus, we are only able to anonymize 43% of the bounding boxes in the validation set. Of the faces that are not anonymized, 22% are partially occluded, and 30% are heavily occluded. For the remaining non-anonymized faces, 70% has a resolution smaller than $14x14$. Note, for each experiment in Table 4.1, we anonymize

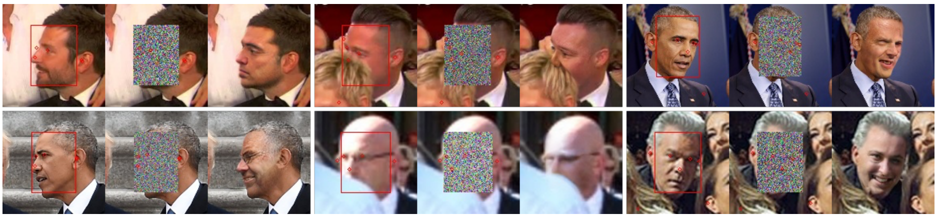


Figure 4.1: **DeepPrivacy results** on a diverse set of images. The left image is the original image annotated with the bounding box and keypoints, the middle image is the input image to our generative model, and the right image is the generated image. Note that our generator never sees any privacy-sensitive information.

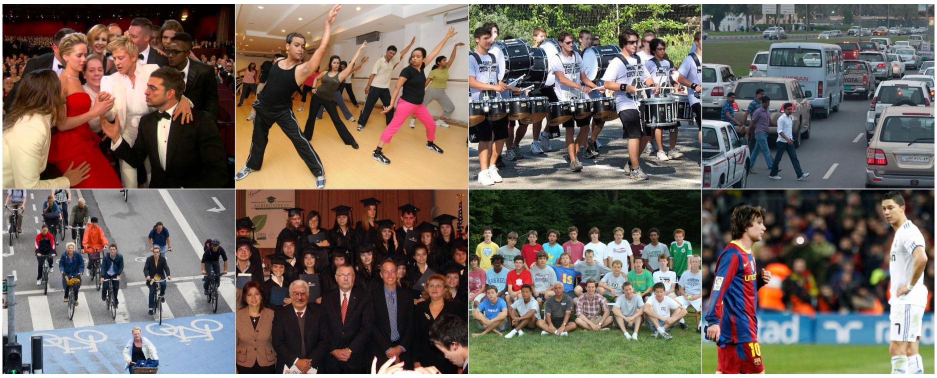


Figure 4.2: **Anonymized images from DeepPrivacy.** Every single face in the images is generated. We recommend the reader to zoom in on the pictures.

the same bounding boxes.

4.3 Experimental Results

In this section, we present and briefly discuss the results of our proposed model. We perform experiments on the FDF dataset and the WIDER-Face dataset. Figure 4.1 shows the original image, the conditional information, and the generated image of DeepPrivacy. Figure 4.2 shows DeepPrivacy on images with several challenges, such as high occlusion and irregular poses. More results can be seen

Anonymization method	Easy	Medium	Hard
No Anonymization	96.6%	95.7%	90.4%
Blacked out	20.6%	29.4%	44.2%
Pixelation (16x16)	95.2%	94.2%	**89.9%**
Pixelation (8x8)	91.0%	91.8%	88.6%
9x9 Gaussian Blur ($\sigma = 3$)	95.1%	92.2%	82.4%
Heavy Blur (filter size = 30% face width)	82.8%	85.5%	85.5%
DeepPrivacy (Ours)	**95.4%**	**94.4%**	89.6%

Table 4.1: **Face detection AP on the WIDER Face [75] validation dataset**. The face detection method used is DSFD [35], the current state-of-the-art on WIDER-Face.

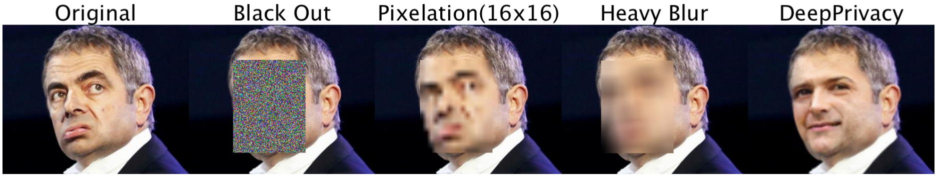


Figure 4.3: **Different anonymization methods** on a face in the WIDER Face validation set.

in Figure 6.1. Also, the fully anonymized WIDER-Face validation set is uploaded to Google Drive [3].

4.3.1 Effect of Anonymization for Face Detection

To evaluate the impact of anonymization, we anonymize the WIDER-Face [75] validation set. We evaluate the AP of a face detection method on the anonymized dataset and compare the results to the original dataset. We report the standard metrics for the different difficulties for the WIDER-Face dataset. Table 4.1 shows the AP of different anonymization techniques. In comparison to the original dataset, DeepPrivacy only degrades the AP by 1.2%, 1.3%, and 0.9% on the easy, medium, and hard difficulties, respectively.

We compare DeepPrivacy anonymization to simpler anonymization methods;

[3] The anonymized WIDER-Face validation dataset can be seen *on Google Drive*. Note that each anonymized face is annotated with a blue bounding box.

Model	FID
With Pose	**2.63**
Without Pose	4.80

(a) Result of using conditional pose.

Discriminator	FID
Deep Discriminator	5.04
Wide Discriminator	**2.63**

(b) Result of the deep and wide discriminator.

#parameters	FID
12M	2.63
46M	**1.53**

(c) Result of different model sizes.

Table 4.2: **Ablation experiments** on our architecture. We report the frèchet inception distance after showing the discriminator 23.4M images (lower is better). For results in Table 4.2a and Table 4.2b, we use a model size of 12M parameters for both the generator and discriminator.

black-out, pixelation, and blurring. Figure 4.3 illustrates the different anonymization methods. DeepPrivacy generally achieves a significant higher AP compared to all other methods, except for $16x16$ pixelation. $8x8$ pixelation suffers significantly in terms of AP. We want to repeat that pixelation and blurring are shown to be insufficient anonymization methods, unable to remove all privacy-sensitive information [16, 49, 50].

WIDER-Face ”hard” of the validation dataset consists of considerably small faces. For the ”easy” challenge, only 43% has a resolution larger than 16×16, and 81.4% has a resolution larger than 8×8. For the ”medium” challenge, 29.9% has a resolution larger than 16×16, and 77.0% has a resolution larger than 8×8 For the ”hard” challenge, 0% has a resolution larger than 16×16, and 23.5% has a resolution larger than 8×8. For any resolution lower than $16x16$, $16x16$ pixelation has no effect. The observant reader might notice that for the ”hard” challenge, $16x16$ pixelation should have no effect; however, the AP is degraded in comparison to the original dataset (see Table 4.1). The only explanation for this behaviour is that pixelating different faces, not present in the ”hard” set, has an effect on detecting faces in the ”hard” set (Note that ”easy” and ”hard” faces can be present in the same image).

4.3.2 Ablation experiments

We perform several ablation experiments to evaluate our architecture choices. We report the average Frèchet Inception Distance between the original image and the anonymized image for each experiment. We calculate FID from a validation set of $50,000$ faces from the FDF dataset. Table 4.2 shows the results of our ablation experiments and we discuss it in detail next. Note, our final model is trained for

42M images and converges to an FID of 1.08.

Effect of pose information: Face poses provided as conditional information improves our model significantly, as seen in Table 4.2a. The FDF dataset has a large variance of faces in all different poses, and we find it necessary to include sparse pose information to generate realistic faces. In contrast, when trained on the CelebA dataset, our model completely ignores the given pose information.

Discriminator Architecture: Table 4.2b compares the quality of images for a deep and wide discriminator. With a deeper network, the discriminator struggles to converge, leading to poor results. We use no normalization layers in neither of these networks, causing deeper networks to suffer from exploding forward passes and vanishing gradients. Even though, Brock *et al.* [8] also observes that a deeper network architecture degrades the overall image quality. Also, we experimented with a discriminator with no modifications to the number of parameters, but this was not able to generate realistic faces.

Model size: We empirically observe that increasing the number of filters in each convolution improves image quality drastically. As seen in Table 4.2c, we train two different models with $12M$ and $46M$ parameters. Unquestionably, increasing the number of parameters generally enhances image quality. For both experiments, we use the same hyperparameters; the only thing changed is the number of filters in each convolution.

We further experimented with a model with $160M$ parameters. We had to reduce the batch size significantly, and we were unable to converge it to an image size of 128×128, due to computational limitations. The preliminary results on 64×64 resolution were slightly worse than our model with $46M$ parameters, probably due to the reduced batch size.

Chapter 5

Discussion

In this chapter, we will discuss the results presented with a focus on our research questions. Then, we will review the limitations of our framework and perform in-depth analysis of our generative model.

5.1 Evaluation

In this section, we will review the research questions presented in Section 1.2 and conclude our findings.

RQ1: What object detection and pose estimation methods are suitable for this application?

Our object detection method is the cornerstone of our model, and the choice of this method is highly dependent on the generative model. We chose DSFD for face detection and Mask R-CNN for pose estimation.

The choice of DSFD [35] for face detection is the perfect balance between performance and inference time. DSFD is state-of-the-art on a wide range of face detection datasets and uses an SSD based approach which improves inference time significantly in comparison to region proposal networks (e.g: TinyFace [22]).

We found Mask R-CNN to be the best-suited model for pose estimation in our case. The vast open-source resources enable us to choose between inference time and keypoint detection performance by changing the CNN backbone. Furthermore, Mask R-CNN is easy to extend for face detection by including an additional output head. However, Mask R-CNN struggles in several scenarios on the WIDER-Face dataset; being able to detect keypoints for only 40% of the present faces, as discussed in Section 4.2.3. For these situations, the faces are either low-resolution or heavily occluded. Alternative pose estimation models (OpenPose or DensePose) would struggle in these scenarios as well. For low-resolution situations, generating a realistic face is more straightforward, and we might not require pose estimation in these situations. An alternative solution could be a secondary generative model that is independent of pose estimation to handle low-resolution situations.

Using two different models for face detection and pose estimation enables us to do the computation in parallel; still, a unified face detection and pose estimation system would improve inference time significantly. Also, training the task in parallel with multi-task learning can help the model to generalize [19]. This is further discussed in Section 6.1

RQ2: How can we generate a realistic human face, fitting for a given situation?

The experiments presented in this thesis reflects that conditional GAN is a suitable method to generate a realistic human face fitting a given situation. On the FDF validation set, the generated images has a significant low FID, indicating that the generated images are similar to the original training dataset. This illustrates that the generated images share very similar features to the original training dataset.

Our method proves its ability to generate objectively high-quality images for a diversity of backgrounds and poses. However, for several faces, a human can easily distinguish a generated face upon closer inspection. Often this is caused by small artifacts in the face. For all observed images, we note that the generative model generates a seamless transition between the generated face and the original background; having no "borders" to the extracted bounding box.

The generated images of our model illustrates its ability to perform complex semantic reasoning. Generating a natural face for a large variety of images is extremely difficult for a generative model. We require the model to perform

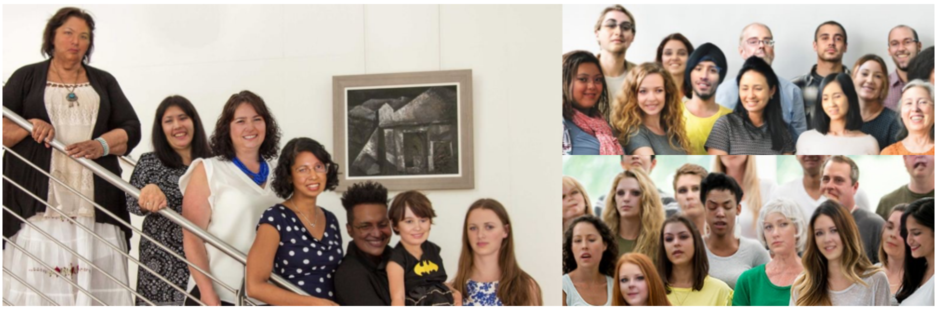


Figure 5.1: **Generalization to unseen images**. DeepPrivacy can handle a diverse set of poses, background clutter, ethnicity, ages, and occlusions. All faces in the images are generated.

complex semantic reasoning to understand the placement of eyes, mouth, and ears while maintaining a seamless transition between the generated face and the original image. Handling this kind of variation is an issue even for non-conditional GANs, where the state-of-the-art on ImageNet is unable to generate consistently realistic images for several classes [8]. For example, Brock *et al.* [8] reports that a model trained on ImageNet is more successful at generating dogs (which makes up a large portion of the ImageNet dataset) than crowds (which comprise a small portion of the dataset and have more large-scale structure).

RQ3: How does the proposed framework perform on real images, which are not present in the training data?

The presented results on the WIDER-Face dataset illustrates that our model can generalize to unseen situations and real-world images [1]. From observing the model on a diverse set of images, we argue that our model can handle a broad collection of facial poses, ethnicities, ages, background clutter, and occlusions. Figure 5.1 shows DeepPrivacy applied to people with different ages, ethnicities and poses. Also, we recommend the reader to take a closer look at the anonymized WIDER-Face dataset presented in Section 4.3.

There exist no perfect quantitative measure to evaluate a model's ability to generalize to unseen images. Still, the Fréchet Inception Distance has shown to be a valuable measurement for this. Our model achieves a small FID for the whole FDF validation dataset. The FDF dataset consists of a large diversity of images,

[1] Note that our model is never trained on any images in the WIDER-Face dataset.

and the low FID reflects our model's ability to handle this kind of variety. We note that the FDF dataset is collected from the Flickr websites and inherently contains the biases of the Flickr dataset. Even though, with validating on the WIDER-Face dataset, we conclude that our model generalizes well to unseen images and we observe no sorts of overfitting on the training set.

RQ4: How can we evaluate the impact of anonymization to further train and validate deep learning models?

Our experiments on the WIDER-Face dataset reflects that anonymized data is suitable for validation of machine learning models. In this thesis, we focus solely on face anonymization, and further work should be done on analyzing the impact on other popular deep learning tasks, such as instance segmentation and general object detection. Also, we perform no experiments focused on training of deep learning models on the anonymized dataset. But, we believe that the experiments shown with validation indicates that the generated image distribution is very similar to the original data distribution; therefore, it is will suited for training.

Evaluating the impact of anonymization on the WIDER-Face validation dataset gives us an indication of the quality of our anonymization, but several factors impact our results. First of all, we are unable to anonymize every bounding box due to Mask R-CNN not being able to detect keypoints for each face. Secondly, the hyperparameters of DSFD [35] are tuned for the original WIDER-Face validation dataset, which might degrade the performance if we slightly change the image.

The results presented in Section 4.3.1 argues that our model is able to retain the original data distribution and reduce the impact of our anonymization. Traditional anonymization tools, such as blurring and pixelation, degrades the performance of the face detection significantly, indicating that these approaches destroy the existing data distribution.

5.2 Limitations

Our method proves its ability to generate objectively good images for a diversity of backgrounds and poses. However, it still struggles in several challenging scenarios. Figure 5.2 illustrates some of these. These issues can impact the generated image quality, but our solution ensures the removal of all privacy-sensitive

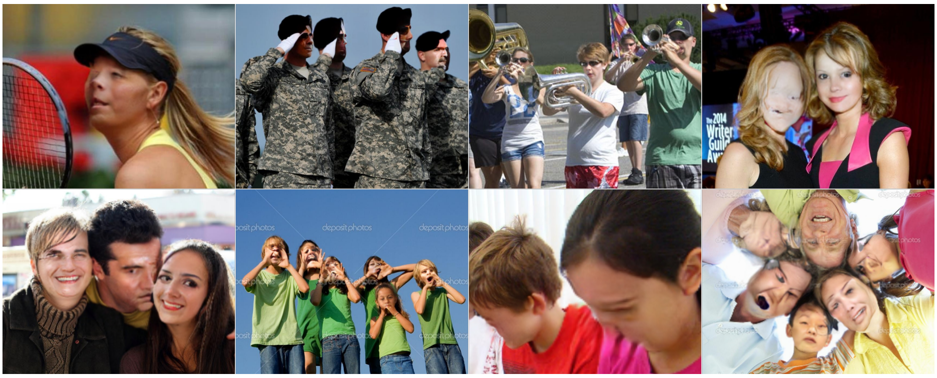


Figure 5.2: **Failure cases of DeepPrivacy.** The four leftmost images indicates that DeepPrivacy struggles in cases of complex backgrounds, and high occlusion of the face. The image in column 3, row 2 has noisy pose annotations, resulting in unrealistic faces. The image in column 4, row 2 illustrates that DeepPrivacy struggles in cases of highly irregular poses. Finally, we notice in some scenarios (column 4, row 1), the image has a perfect pose annotation; however, our generative model is unable to generate a realistic face, even though it is a fairly simple scenario.

information. These limitations can be a limitation of our dataset, as we have collected human faces mainly from traffic and cities.

Faces occluded with high fidelity objects are challenging when generating a realistic face. For example, in Figure 5.2 several images have persons covering their face with hands. To generate a face in this scenario requires complex semantic reasoning, which is still a difficult challenge for GAN.

Handling non-traditional poses can cause our model to generate corrupted faces. We use a sparse pose estimation to describe the facial pose, but there is no limitation in our architecture to include a dense pose estimation. A denser pose estimation would, most likely, improve the performance of our model in cases of irregular poses. However, predicting a denser pose estimation is more challenging and would restrict the practical use case of our method.

5.3 Further Analysis

From an in-depth analysis of our network, we present several interesting findings working with conditional GANs using progressive growing. We will discuss these in detail next.

Training Stability and Convergence

Training stability is a substantial issue for GANs and can impact final image quality. We include several techniques to improve the stability of our network, especially Progressive Growing of GANs, but we still observe significant oscillation in the wasserstein loss during training. This instability is prominent during transition phases when we scale the resolution of our model, as seen in Figure 5.3. Even though, our network can successfully train for a large set of hyperparameter choices.

The gradient penalty function is a large cause of instability during the transition phase. As seen in Equation 2.3, the gradient penalty is calculating the norm of the gradient on the input image. Therefore, the scale of the gradient penalty is dependent on the image resolution, and it grows by a factor of 2^2 each time we increase the image resolution. To counteract this, we experimented with linearly scaling the gradient penalty to mirror the increase of the gradient penalty. However, linearly scaling the gradient penalty during the transition phase is suboptimal, as this would remove the K-Lipschitz constraint on our discriminator.

Deterministic Output of Generator

We analyze the sensitivy to the input of our generator. Our generator has two inputs; conditional information (context and pose), and a latent variable z drawn form a normal distribution ($\mathcal{N}(0,1)$). To achieve a Nash Equilibrium, the z vector is necesarry and the generator should be dependent on z [14]. As seen in Figure 5.4, our generator is close to independent on z. This can cause suboptimal training of our networks. Therefore, we experiment with "forcing" the generator to care about the variable. We introduced an additional L_2 loss to make the generator reconstruct z at the end of the network. However, we observed insignficant difference in the dependency of the latent variable z, and the resulting FID was the same.

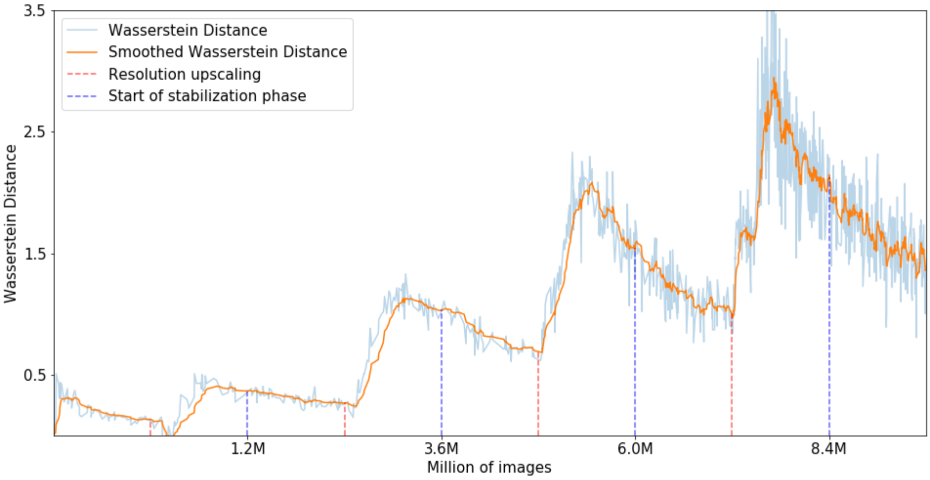


Figure 5.3: **Instability of Wasserstein Distance** during training. The network was trained to 64×64 resolution, with 600K images for transition and stabilization faces. The training was stopped after 9.6M images.

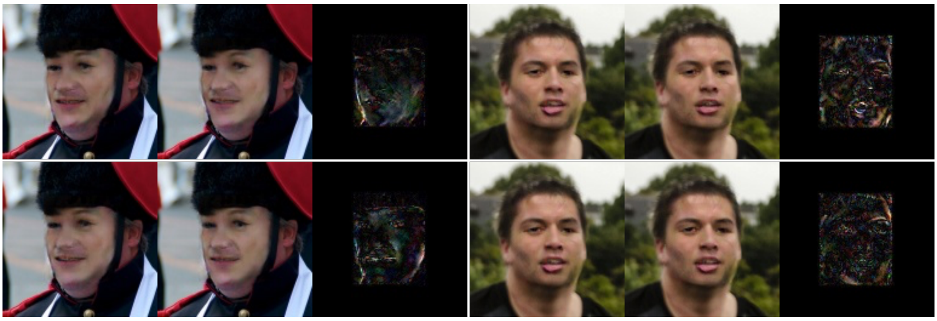


Figure 5.4: **Deterministic Generator Output.** By changing our latent variable z, we notice minimal difference in generated images. The two left images has the same conditional information with different z vector. The right image is the L_1 distance per pixel between the images, normalized between 0-1. We recommend the reader to zoom in to notice minor differences.

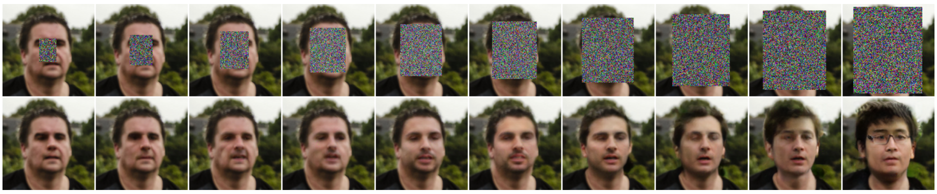


Figure 5.5: **Bounding box sensitivity** experiment on our our generative model. DeepPrivacy is robust to small adjustments in the bounding box.

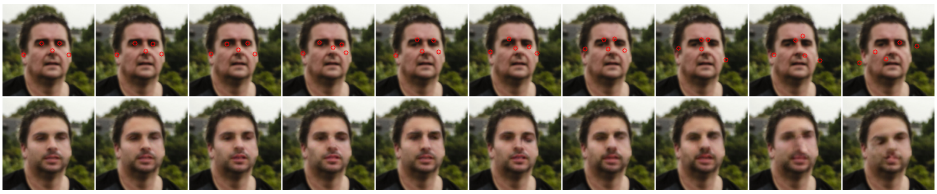


Figure 5.6: **Pose information sensitivity** experiment on our generative model. *Leftmost* image is the original pose information. *Rightmost* image is the pose information with random noise within $[-0.1 \cdot \text{image width}, 0.1 \cdot \text{image width}]$.

A deterministic output of the generator in a conditional GAN has been observed by several authors [24]. Previous methods has used dropout to force a non-deterministic generator [24]; however, we experimented with this and observed significantly worse image quality. Due to the enourmous size of our dataset, we believe the deterministic output of our generator does not harm training. Still, further work into analyzing the cause of this, and the impact on final results should be performed.

Sensitivity to Detection Models

For practical use, we desire our generative model to be independent on the detection system, such that the generative model is robust to small variations in the bounding box or pose annotation. Figure 5.5 shows our generative model on the same image for various bounding boxes. Notice that our generative model is robust to major changes in the bounding box. Figure 5.6 shows our generative model on the same image for multiple pose annotations. Our generative model is robust to minor adjustments in pose information, but the more non-traditional poses generates unrealistic images. From these experiments and overall quality of generated images on the FDF dataset, our model is robust to rather large

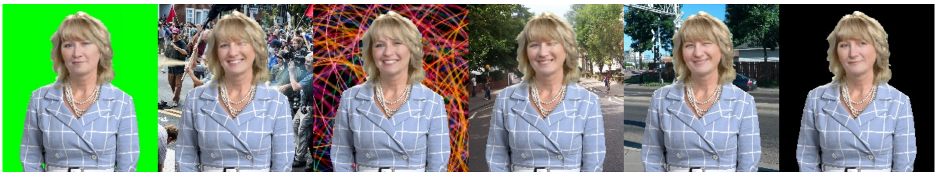


Figure 5.7: **Background sensitivity experiment** on our generative model. We take greenscreen image with a person, and anonymize the image with different image background. Each face here is generated.

changes in annotations.

Robustness to Image Background

We analyse the sensitivity of our generative model to different background information. Figure 5.7 shows the same person for different backgrounds, where we took an original image with greenscreen to change the background of the face. Notice that our model is able to generate realistic faces for a large variety of backgrounds. Also, an interesting observation is that the personal traits of the individual is similar for all the different backgrounds. This indicates that DeepPrivacy generates the identity dependent on the given information about the person, and not the background information.

Chapter 6

Conclusion and Further Work

We propose a multi-stage pipeline to automatically anonymize faces in images without destroying the original data distribution. The presented results on the WIDER-Face dataset reflects our model's ability to generate high-quality images. Also, the diversity of images in the WIDER-Face dataset shows the practical applicability of our model. The current state-of-the-art face detection method can achieve 98.8% of the original AP on the anonymized WIDER-Face validation set. In comparison to previous solutions, this is a significant improvement to both the generated image quality, and the certainty of anonymization. Furthermore, the presented ablation experiments on the FDF dataset suggests that our model improves with an increased number of parameters. Also, inclusion of sparse pose information is necessary to generate high-quality images.

Our generative model is a conceptually simple generative adversarial network, easily extendable for further improvements. Handling irregular poses, difficult occlusions, complex backgrounds, and temporal consistency in videos are still subjects for further work. We believe our contribution will be an inspiration for further work into ensuring privacy in visual data.

6.1 Future Work

The presented results are encouraging, but it requires further work to ensure robustness to a large diversity of scenarios. In this section, we will discuss alternative objective functions for improved generated image quality and training efficiency. Also, we will briefly discuss our proposed FDF dataset and techniques to unify object detection and pose estimation into a single model for improved inference time. Finally, we review the ability to extend our proposed model to the video domain.

Alternative Objective Functions

Even though we use progressive growing throughout training, we experience significant instability in our network during transition phases, as discussed in Section 5.3. Gradient penalty is sub-optimal for progressive growing GANs, as it scales quadratically with the resolution. Other objective functions can be beneficial for the stability of the network. StyleGAN [30] argues that the non-saturating loss with R-regularization improves final image quality; however, the R-regularization term also scales quadratically with the resolution. Miyato *et al.* [47] proposes Spectral Normalization to enforce the K-Lipschitz criteria instead of gradient penalty, and this would not be dependent on the image resolution. Replacing gradient penalty with spectral normalization can be an area worth exploring to improve training stability.

FDF Dataset

FDF is a crucial part of our model's success, giving us a dataset with a diverse set of faces. However, the dataset is unnecessarily large (1.3M images), making it computationally expensive to use for training. A smaller dataset, but representative of the data distribution, would reduce the requirement of computational power and improve training time. Active Learning [61] tries to minimize the amount of data required to train a machine learning model. In active learning, the key idea is to extract images that are unique and representative for the data distribution while removing any images that do not provide new knowledge. To pick new samples for a dataset, we often look at the classification certainty of a trained model; however, this is not a trivial task for generative models. Effectively selecting a representative subset of the FDF dataset could limit the computational power requirements, and it is shown that active learning can improve convergence time [61].

Unified Object Detection and Pose Estimation

Our pipeline does face detection and pose estimation in parallel with two different models. For the scope of this thesis, this parallel pipeline is a suitable solution. However, similar features have to be computed separately for each model, which is very computational inefficient. Alternatives to this could be to extend the Mask R-CNN network with an additional output head to include a face detection branch. The modifications to the model are minimal, but the dataset requirement is a substantial issue to make this work. We would require a dataset with an object bounding box, face bounding box, segmentation masks, and keypoint annotations. The COCO dataset [38], which Mask R-CNN is originally trained for, includes these annotations except for bounding boxes of faces. Another option to remove the dataset constriction could be to transfer learn a pre-trained Mask R-CNN on a face detection datasets, such as WIDER-Face.

Dense Pose Information

Our generative model uses sparse pose information about the location of the eyes, ears, nose, and shoulders. The presented ablation experiments indicate that sparse pose information improves overall image quality significantly. Using a denser pose estimation to guide the generative model is a promising area for further work. DensePose [2] maps each pixel of the human body to a 3d body part, giving us a dense pose estimation. From this, we would be able to extract a semantic segmentation of the face region instead of a rectangular bounding box. However, our proposed model does not scale well with dense pose estimation. We encode the pose information as $K \times M \times M$ one-hot encoded image, where K is the number of keypoints and M is the image width and height. With dense pose information the memory requirement would be infeasible. To support a dense pose estimation, we require an alternative approach to input the pose information.

Temporal Consistency for Videos

The work done in this thesis is focused on the image domain, anonymizing images frame-by-frame and ignoring any temporal consistency. Applying our model on a video generates realistic faces, but introduces significant flickering and a human notices inconsistency over time [1]. Several works have focused on using GANs for generating videos with temporal consistency; two popular tasks are style transfer for videos [11, 18, 58], and video-to-video synthesis [70]. Wang *et al.* [70] proposes a video-to-video GAN, where they use a basic markov assumption to enforce temporal consistency; that is, a single frame is dependent on the previous two frames. This markov assumption performs well to enforce short temporal consistency, but they discuss the limitation of this assumption to generate coherent synthesis with long temporal consistency.

[1] Example video can be downloaded from Google Drive. ”detected.mp4” is the original video with detections, ”generated.mp4” is the final result, and ”marked.mp4” is the generated video with annotations indicating which faces are anonymized.

Bibliography

[1] DeepFake: Face Swap. `https://github.com/deepfakes/faceswap`, 2018.

[2] Rıza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. Densepose: Dense human pose estimation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7297–7306, 2018.

[3] Riza Alp Guler, George Trigeorgis, Epameinondas Antonakos, Patrick Snape, Stefanos Zafeiriou, and Iasonas Kokkinos. Densereg: Fully convolutional dense shape regression in-the-wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6799–6808, 2017.

[4] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele.

[5] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

[6] Shane Barratt and Rishi Sharma. A note on the inception score. *arXiv preprint arXiv:1801.01973*, 2018.

[7] Michael Boyle, Christopher Edwards, and Saul Greenberg. The effects of filtered video on awareness and privacy. In *Proceedings of the 2000 ACM conference on Computer supported cooperative work*, pages 1–10. ACM, 2000.

[8] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

[9] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields. In *arXiv preprint arXiv:1812.08008*, 2018.

[10] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017.

[11] Dongdong Chen, Jing Liao, Lu Yuan, Nenghai Yu, and Gang Hua. Coherent online video style transfer. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1105–1114, 2017.

[12] Ross Girshick, Ilija Radosavovic, Georgia Gkioxari, Piotr Dollár, and Kaiming He. Detectron. `https://github.com/facebookresearch/detectron`, 2018.

[13] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[15] Ralph Gross, Latanya Sweeney, Jeffrey Cohn, Fernando De la Torre, and Simon Baker. Face de-identification. In *Protecting privacy in video surveillance*, pages 129–146. Springer, 2009.

[16] Ralph Gross, Latanya Sweeney, Fernando De la Torre, and Simon Baker. Model-based face de-identification. In *2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06)*, page 161. IEEE, 2006.

[17] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, pages 5767–5777, 2017.

[18] Agrim Gupta, Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Characterizing and improving stability in neural style transfer. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4067–4076, 2017.

[19] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[21] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, pages 6626–6637, 2017.

[22] Peiyun Hu and Deva Ramanan. Finding tiny faces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 951–959, 2017.

[23] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1501–1510, 2017.

[24] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[25] Vidit Jain and Erik Learned-Miller. Fddb: A benchmark for face detection in unconstrained settings. Technical Report UM-CS-2010-009, University of Massachusetts, Amherst, 2010.

[26] Youngjoo Jo and Jongyoul Park. Sc-fegan: Face editing generative adversarial network with user's sketch and color. *arXiv preprint arXiv:1902.06838*, 2019.

[27] Alexia Jolicoeur-Martineau. The relativistic discriminator: a key element missing from standard gan. *arXiv preprint arXiv:1807.00734*, 2018.

[28] Amin Jourabloo, Xi Yin, and Xiaoming Liu. Attribute preserved face de-identification. *Proceedings of 2015 International Conference on Biometrics, ICB 2015*, pages 278–285, 2015.

[29] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

[30] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *arXiv preprint arXiv:1812.04948*, 2018.

[31] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[32] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[33] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. *arXiv preprint arXiv:1904.06991*, 2019.

[34] Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*.

[35] Jian Li, Yabiao Wang, Changan Wang, Ying Tai, Jianjun Qian, Jian Yang, Chengjie Wang, Jilin Li, and Feiyue Huang. Dsfd: dual shot face detector. *arXiv preprint arXiv:1810.10220*, 2018.

[36] Yijun Li, Sifei Liu, Jimei Yang, and Ming-Hsuan Yang. Generative face completion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3911–3919, 2017.

[37] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2117–2125, 2017.

[38] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[39] Guilin Liu, Fitsum A Reda, Kevin J Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 85–100, 2018.

[40] Ming-Yu Liu, Xun Huang, Arun Mallya, Tero Karras, Timo Aila, Jaakko Lehtinen, and Jan Kautz. Few-shot unsupervised image-to-image translation, 2019.

[41] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[42] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

[43] Mario Lucic, Karol Kurach, Marcin Michalski, Sylvain Gelly, and Olivier Bousquet. Are gans created equal? a large-scale study. In *Advances in neural information processing systems*, pages 700–709, 2018.

[44] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2794–2802, 2017.

[45] Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, et al. Mixed precision training. *arXiv preprint arXiv:1710.03740*, 2017.

[46] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[47] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

[48] Mahyar Najibi, Pouya Samangouei, Rama Chellappa, and Larry S Davis. Ssh: Single stage headless face detector. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4875–4884, 2017.

[49] Carman Neustaedter, Saul Greenberg, and Michael Boyle. Blur filtration fails to preserve privacy for home-based video conferencing. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 13(1):1–36, 2006.

[50] Elaine M Newton, Latanya Sweeney, and Bradley Malin. Preserving privacy by de-identifying face images. *IEEE transactions on Knowledge and Data Engineering*, 17(2):232–243, 2005.

[51] NVIDIA. A pytorch extension: Tools for easy mixed precision and distributed training in pytorch. `https://github.com/NVIDIA/apex`, 2019.

[52] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2642–2651. JMLR. org, 2017.

[53] Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. *arXiv preprint arXiv:1601.06759*, 2016.

[54] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[55] Zhongzheng Ren, Yong Jae Lee, and Michael S Ryoo. Learning to anonymize faces for privacy preserving action detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 620–636, 2018.

[56] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[57] Kevin Roth, Aurelien Lucchi, Sebastian Nowozin, and Thomas Hofmann. Stabilizing training of generative adversarial networks through regularization. In *Advances in neural information processing systems*, pages 2018–2028, 2017.

[58] Manuel Ruder, Alexey Dosovitskiy, and Thomas Brox. Artistic style transfer for videos. In *German Conference on Pattern Recognition*, pages 26–36. Springer, 2016.

[59] Mehdi SM Sajjadi, Olivier Bachem, Mario Lucic, Olivier Bousquet, and Sylvain Gelly. Assessing generative models via precision and recall. In *Advances in Neural Information Processing Systems*, pages 5228–5237, 2018.

[60] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in neural information processing systems*, pages 2234–2242, 2016.

[61] Burr Settles. Active learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2009.

[62] Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *CVPR*, 2017.

[63] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[64] Supasorn Suwajanakorn, Steven M Seitz, and Ira Kemelmacher-Shlizerman. Synthesizing obama: learning lip sync from audio. *ACM Transactions on Graphics (TOG)*, 36(4):95, 2017.

[65] Latanya Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, 2002.

[66] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[67] Justus Thies, Michael Zollhofer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. Face2face: Real-time face capture and reenactment of rgb videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2387–2395, 2016.

[68] Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. YFCC100M: The new data in multimedia research. *arXiv preprint arXiv:1503.01817*, 2015.

[69] Paul Viola and Michael J Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004.

[70] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Video-to-video synthesis. *CoRR*, abs/1808.06601, 2018.

[71] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8798–8807, 2018.

[72] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, volume 2, pages 1398–1402. Ieee, 2003.

[73] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *CVPR*, 2016.

[74] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017.

[75] Shuo Yang, Ping Luo, Chen Change Loy, and Xiaoou Tang. Wider face: A face detection benchmark. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[76] Yasin Yazıcı, Chuan-Sheng Foo, Stefan Winkler, Kim-Hui Yap, Georgios Piliouras, and Vijay Chandrasekhar. The unusual effectiveness of averaging in gan training. *arXiv preprint arXiv:1806.04498*, 2018.

[77] Raymond A Yeh, Chen Chen, Teck Yian Lim, Alexander G Schwing, Mark Hasegawa-Johnson, and Minh N Do. Semantic image inpainting with deep generative models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5485–5493, 2017.

[78] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

[79] Stefanos Zafeiriou, Cha Zhang, and Zhengyou Zhang. A survey on face detection in the wild: past, present and future. *Computer Vision and Image Understanding*, 138:1–24, 2015.

[80] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. *arXiv preprint arXiv:1805.08318*, 2018.

[81] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5907–5915, 2017.

[82] Shifeng Zhang, Xiangyu Zhu, Zhen Lei, Hailin Shi, Xiaobo Wang, and Stan Z Li. S3fd: Single shot scale-invariant face detector. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 192–201, 2017.

[83] C Lawrence Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In *European conference on computer vision*, pages 391–405. Springer, 2014.

Appendices

A. Discriminator Architecture

Discriminator	Act	Output shape	Params
Input image + conditional Info	–	$6 \times 128 \times 128$	–
Conv 1×1	LReLU	$176 \times 128 \times 128$	1.2K
Stack pose information	–	$183 \times 128 \times 128$	–
Conv 3×3	LReLU	$176 \times 128 \times 128$	290K
Conv 3×3	LReLU	$360 \times 128 \times 128$	571K
Downsample	–	$360 \times 64 \times 64$	–
Stack pose information	–	$367 \times 64 \times 64$	–
Conv 3×3	LReLU	$360 \times 64 \times 64$	1.2M
Conv 3×3	LReLU	$720 \times 64 \times 64$	2.3M
Downsample	–	$720 \times 32 \times 32$	–
Stack pose information	–	$727 \times 32 \times 32$	–
Conv 3×3	LReLU	$720 \times 32 \times 32$	4.7M
Conv 3×3	LReLU	$720 \times 32 \times 32$	4.7M
Downsample	–	$720 \times 16 \times 16$	–
Stack pose information	–	$727 \times 16 \times 16$	–
Conv 3×3	LReLU	$720 \times 16 \times 16$	4.7M
Conv 3×3	LReLU	$720 \times 16 \times 16$	4.7M
Downsample	–	$720 \times 8 \times 8$	–
Stack pose information	–	$519 \times 8 \times 8$	–
Conv 3×3	LReLU	$720 \times 8 \times 8$	4.7M
Conv 3×3	LReLU	$720 \times 8 \times 8$	4.7M
Downsample	–	$720 \times 8 \times 8$	–
Stack pose information	–	$727 \times 4 \times 4$	–
Conv 3×3	LReLU	$720 \times 4 \times 4$	4.7M
Conv 4×4	LReLU	$720 \times 1 \times 1$	8.3M
Fully-connected	linear	$1 \times 1 \times 1$	721
Total trainable parameters			**45.5M**

Table 6.1: Discriminator that we use to generate 128x128 images.

B. Additional Generated Images

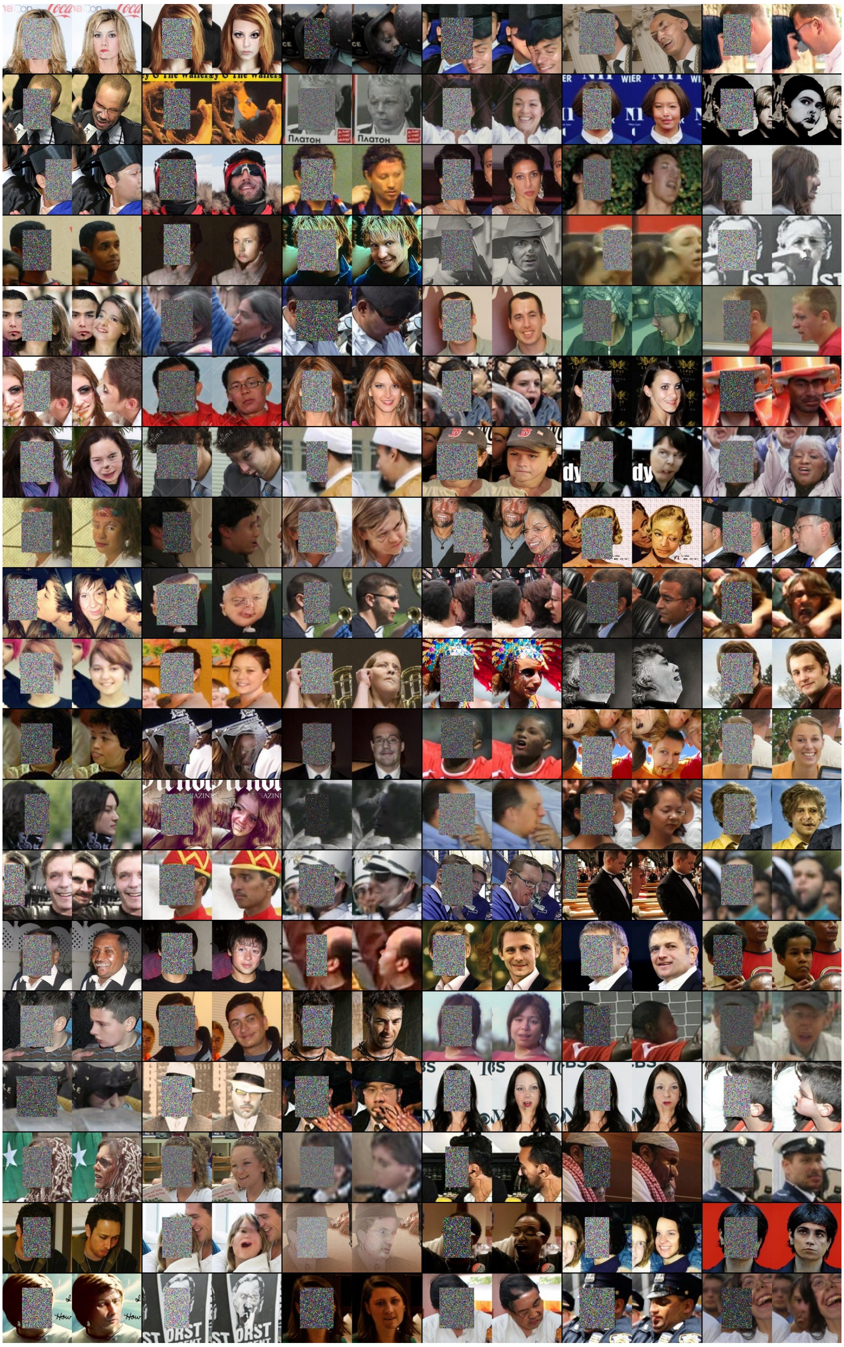


Figure 6.1: **DeepPrivacy results** on a diverse set of faces in the WIDER-Face dataset. The left image is the original image annotated with the bounding box and keypoints, and the right image is the generated image. Note that these faces are randomly picked. We recommend the reader to zoom in.

C. BMVC Paper

This thesis was submitted as a 9-page paper to the British Machine Vision Conference 2019, and is currently under review. Acceptance notification will be sent out by Monday 24th of June, 2019.

DeepPrivacy: A generative adversarial network for face anonymization

BMVC 2019 Submission # 1068

Abstract

We propose a novel architecture which is able to automatically anonymize faces in images while retaining the original data distribution. We ensure total anonymization of all individuals in an image by generating images exclusively on privacy-safe information. Our model is based on a conditional generative adversarial network, generating images considering the original pose and image background. The conditional information enables us to generate highly realistic faces with a seamless transition between the generated face and the existing background. Furthermore, we introduce a diverse dataset of human faces including unconventional poses, occluded faces, and a vast variability in backgrounds. Finally, we present experimental results reflecting the capability of our model to anonymize images while preserving the data distribution, making the data suitable for further training of deep learning models. As far as we know, no other solution has been proposed that guarantees the anonymization of faces while generating realistic images.

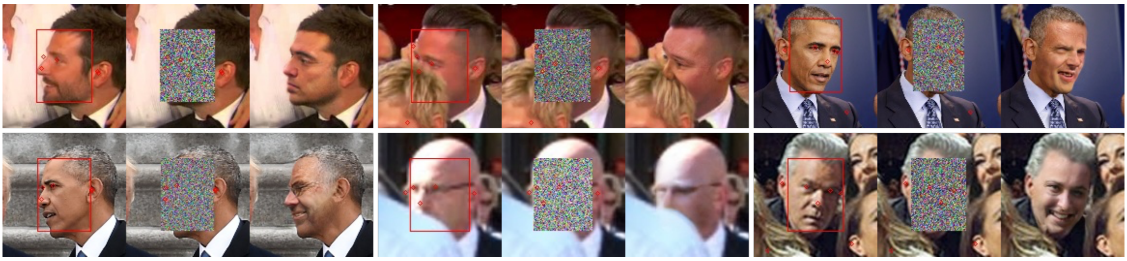


Figure 1: **DeepPrivacy results** on a diverse set of images. The left image is the original image annotated with bounding box and keypoints, the middle image is the input image, and the right image is the generated image. Note that our generator never sees any privacy-sensitive information.

1 Introduction

Privacy-preserving data-processing is becoming more critical every year; however, no suitable solution has been found to anonymize images without degrading the image quality. The General Data Protection Regulation (GDPR) came to effect as of 25th of May, 2018, affecting all processing of personal data across Europe. GDPR requires regular consent from the individual for any use of their personal data. However, if the data does not allow to identify an individual, companies are free to use the data without consent. To effectively anonymize images, we require a robust and highly effective generative model to replace the original face, without destroying the existing data distribution; that is: the output should be a realistic face fitting into the situation.

Anonymizing images while retaining the original distribution is a challenging task. The model is required to remove all privacy-sensitive information, generate a highly realistic face, and the transition between original and anonymized parts has to be seamless. This requires a model that can perform complex semantic reasoning to generate a new anonymized face. For practical use, we desire the model to be able to manage a broad diversity of images, poses, backgrounds, and different persons. Our proposed solution can successfully anonymize images in a large variety of cases, and create realistic faces to the given conditional information.

Our proposed model, called *DeepPrivacy*, is a conditional generative adversarial network [18]. Our generator considers the existing background and a sparse pose annotation to generate realistic anonymized faces. The generator has a U-net architecture [23], which we train with a progressive growing training technique [12] from a starting resolution of 8×8 to 128×128; this substantially improves the final image quality and overall training time. Our generator never receives any privacy-sensitive information, thus ensuring that the generated images are completely anonymized.

For practical use, we assume no demanding requirements for the object and keypoint detection algorithm. Our model requires two simple annotations of the face: (1) a tight bounding box annotation to identify the privacy-sensitive area, and (2) a sparse pose estimation of the face, containing keypoints for the ears, eyes, nose, and shoulders; in total seven keypoints. This keypoint annotation is identical to what Mask R-CNN [6] provides.

We provide a new dataset of human faces, *Flickr Diverse Faces* (FDF), which consists of 1.3M faces with a bounding box and keypoint annotation for each face. This dataset covers a considerably large diversity of facial poses, partial occlusions, complex backgrounds, and different persons. We will make this dataset publicly available along with our source code and pre-trained networks.

We evaluate our model by performing an extensive qualitative and quantitative study of the model's ability to retain the original data distribution. We anonymize the validation set of the WIDER-Face dataset [27], then run a face detection on the anonymized images. The current state-of-the-art, DSFD [14], achieves 98.8% (95.4% out of 96.6% average precision), 98.6% (94,4%/95,7%), and 99.1% (89,6%/90,4%) of the original average precision on the easy, medium, and hard difficulty, respectively; on average, it keeps 98.8% of the original performance. In contrast, 8*x*8 pixelation achieves 96.0%, heavy blur 89.9%, and black-out 33.6% of the original performance. Additionally, we present several ablation experiments that reflect the importance of a large model size and conditional pose information to generate high-quality faces.

In summary, we make the following contributions:

- We propose a novel generator architecture to anonymize faces, which ensures 100% removal of privacy-sensitive information in the original image. The generator can generate realistic looking faces that have a seamless transition to the existing background for various sets of poses and contexts.

- We provide the FDF dataset, including 1.3M faces with a tight bounding box and keypoint annotation for each face. The dataset covers a considerably larger diversity of faces compared to previous datasets.

2 Related Work

De-identifying faces: Currently, there exists a limited number of research studies on the task of removing privacy-sensitive information from an image including a face. Typically, the approach chosen is to alter the original image such that we remove all the privacy-sensitive information. These algorithms can be applied to all images; however, we have no assurance that we remove the privacy-sensitive information. Naive methods that apply simple image distortion have been discussed numerous times in literature [1, 4, 5, 19, 20], such as pixelation and blurring; but, they are inadequate for removing the privacy-sensitive information [4, 19, 20], and they alter the data distribution substantially.

K-same family of algorithms [4, 11, 20] implements the k-anonymity algorithm [25] for face images. Newton *et al*. prove that the k-same algorithm can remove all privacy-sensitive information; but, the resulting images often contain "ghosting" artifacts due to small alignment errors[4].

Generative Adversarial Networks(GANs) [3] has been a highly successful training architecture to model a natural image distribution. GANs enables us to generate new images, often indistinguishable from the real data distribution. It has a broad diversity of application areas, from general image generation [2, 12, 13, 31], text-to-photo generation [30], style transfer [8, 24] and much more. With the numerous contributions since its conception, it has gone from a beautiful theoretical idea to a tool we can apply for practical use cases. In our work, we show that GANs is an efficient tool to remove privacy-sensitive information without destroying the original image quality.

Ren *et al*. [22] look at the task of anonymizing video data by using a generative adversarial network. They perform anonymization by altering each pixel in the original image to hide the identity of the individuals. In contrast to their method, we can ensure the removal of all privacy-sensitive information, as our generative model never observes the original face.

Progressive growing of GANs [12] propose a novel training technique to generate faces progressively, starting from a resolution of 4*x*4 and step-wise increasing it to 1024*x*1024. This training technique improves the final image quality and overall training time. Our proposed model uses the same training technique; however, we perform several alterations to their original model to convert it to a conditional GAN. With these alterations, we can include conditional information about the context and pose of the face. Our final generator architecture is similar to the one proposed by Isola *et al*. [9], but we introduce conditional information in several stages.

Image Inpainting is a closely related task to what we are trying to solve, and it is a widely researched area for generative models [10, 15, 17, 29]. Several research studies have looked at the task of face completion with a generative adversarial network [15, 29]. They mask a specific part of the face and try to complete this part with the conditional information given. From our knowledge, and the qualitative experiments they present in their papers, they are not able to mask a large enough part to remove all privacy-sensitive information. As the masked region grows, it requires a more advanced generative model that understands complex semantic reasoning, making the task considerably harder. In comparison to these methods, we expect a rectangular shaped mask to identify the area we want to inpaint, and we use a more complex dataset than the typical CelebA dataset [17].

Jourabloo *et al*. [11] look at the task of de-identification grayscale images while preserving a large set of facial attributes. This is different from our work, as we do not directly train our generative model to generate faces with similar attributes to the original image. In contrast, our model is able to perform complex semantic reasoning to generate a face that is coherent with the overall context information given to the network, yielding a highly realistic resulting face.

3 The Flickr Diverse Faces dataset

FDF (Flickr Diverse Faces) is a new dataset of human faces, crawled from the YFCC-100M dataset [26]. It consists of 1.3M human faces with a minimum resolution of 128*x*128, containing facial landmarks and a bounding box annotation for each face. The dataset has a vast diversity in terms of age, ethnicity, facial pose, image background, and face occlusion. The dataset was mainly extracted from scenes related to traffic, sports events, and outside activities. In comparison to the FFHQ dataset [13], our dataset is largely more diverse in facial poses, and it is generally much larger; however, the FFHQ dataset has a higher resolution.

The FDF dataset is a high-quality dataset with few annotation errors. The faces are automatically labeled with state-of-the-art keypoint and bounding box models, and we use a high confidence threshold for both the keypoint and bounding box predictions. The faces are extracted from $964,099$ images in the YFCC100-M dataset. For keypoint estimation, we use Mask R-CNN [6], with a ResNet-50 FPN backbone [16]. For bounding box annotation, we use the Single Shot Scale-invariant Face Detector [32]. Each keypoint is matched with a bounding box if the eye and nose annotation are within the tight bounding box. Each bounding box and keypoint has a single match, and we match them with a greedy approach based on descending prediction confidence.

4 Model

Our proposed model is a conditional GAN, generating images based on the surrounding of the face and sparse pose information. Figure 1 shows the conditional information given to our network. We base our model on the one proposed by Karras *et al*. [12]. Their model is a non-conditional GAN, and we perform several alterations to include conditional information.

We use seven facial landmarks to describe the pose of the face, including the following keypoints: left/right eye, left/right ear, left/right shoulder, and nose. To reduce the number of parameters in the network, we pre-process the pose information into a one-hot encoded image of size $K \times M \times M$, where K is the number of landmarks and M is the target resolution.

Progressive growing training technique improves the final image quality and overall training time, and it's crucial for our model's success. We apply progressive growing to both the generator and discriminator to grow the networks from a starting resolution of 8. We double the resolution each time we expand our network until we reach the final resolution of 128×128. To include the pose information through the whole training period, we decide to include this information for each resolution for both the discriminator and generator.

4.1 Generator Architecture

Figure 2 shows our proposed generator architecture for 128×128 resolution. Our generator has a U-net[23] architecture to include background information. The encoder and decoder have the same number of filters in each convolution, but the decoder has an additional 1×1 bottleneck convolution after each skip connection. This bottleneck design reduces the number of parameters in the decoder significantly. To include the pose information for each

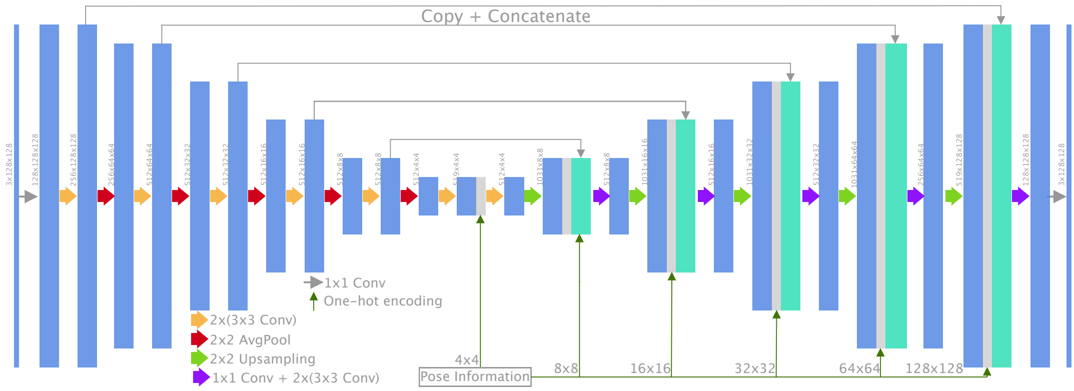


Figure 2: **Our generator architecture** for 128*x*128 resolution. Each convolutional layer is followed by pixel normalization [12] and LeakyReLU($\alpha = 0.2$). After each upsampling layer, we concatenate the upsampled output with pose information and the corresponding skip connection.

resolution, we concatenate the output after each upsampling layer with pose information and the corresponding skip connection. The general layer structure is identical to Karras *et al.* [12], where we use pixel replication for upsampling, pixel normalization and LeakyReLU after each convolution, and equalized learning rate instead of careful weight initialization.

Progressive growing: Each time we increase the resolution of the generator, we add two 3×3 convolutions to the start of the encoder and the end of the decoder. We use a transition phase identical to Karras *et al.* [12] for both of these new blocks, making the network stable throughout the training. We note that the network is still unstable during the transition phase, but it is significantly better compared to training without progressive growing.

4.2 Discriminator Architecture

Our proposed discriminator architecture is identical to the one proposed by Karras *et al.* [12], with a few exceptions. First, we include the background information as conditional input to the start of the discriminator, making the input image have six channels instead of three. Secondly, we include pose information at each resolution of the discriminator. The pose information is concatenated with the output of each downsampling layer, similarly to the decoder in the generator. Finally, we remove the mini-batch standard deviation layer presented by Karras *et al.* [12], as we find the diversity of our generated faces satisfactory.

The adjustments made to the generator doubles the number of total parameters in the network. To follow the design lines of Karras *et al.* [12], we desire that the complexity in terms of the number of parameters to be similar for the discriminator and generator. We evaluate two different discriminator models, which we will name the *deep discriminator* and the *wide discriminator*. The deep discriminator doubles the number of convolutional layers for each resolution. To mimic the skip-connections in the generator, we wrap the convolutions for each resolution in residual blocks. The wider discriminator keeps the same architecture; however, we increase the number of filters in each convolutional layer by a factor of $\sqrt{2}$.

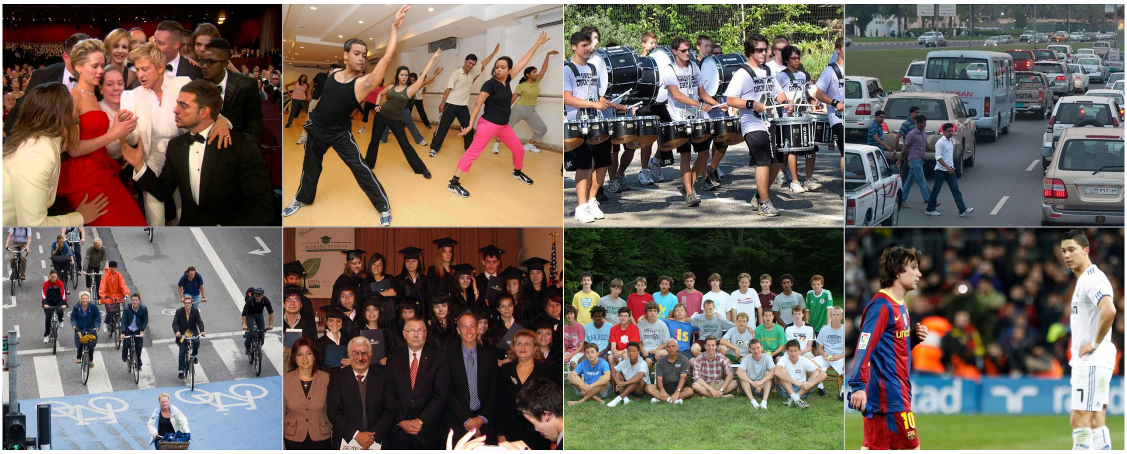


Figure 3: Anonymized images from DeepPrivacy. Every single face in the images has been generated.

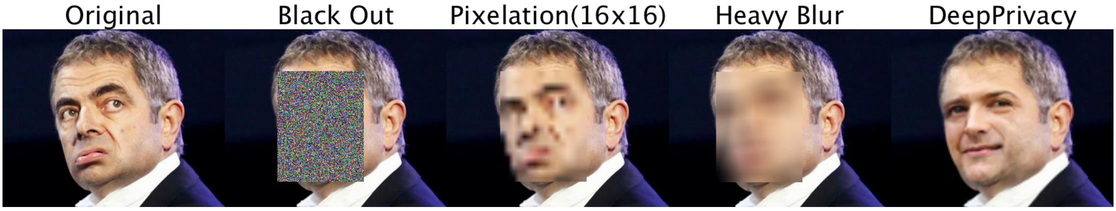


Figure 4: **Different anonymization methods** on a face in the WIDER Face validation set.

5 Experiments

DeepPrivacy can robustly generate anonymized faces for a vast diversity of poses, backgrounds, and different persons. From qualitative evaluations of our generated results on the WIDER-Face dataset [27], we find our proposed solution to be robust to a broad diversity of images. Figure 3 shows several results of our proposed solution on the WIDER-Face dataset. Note, the network is trained on the FDF dataset; we do not train on any of the images in the WIDER-Face dataset. With our paper, we will release the full WIDER-Face validation set anonymized by DeepPrivacy. (For this submission, we deliver a randomly picked subset of this dataset, as the total file limit is 100MB.)

We evaluate the impact of anonymization on the WIDER-Face [27] dataset. We measure the average precision of a face detection method on an anonymized dataset and compare this to the original dataset. We report the standard metrics for the different difficulties for the WIDER-Face dataset. Additionally, we perform several ablation experiments on our proposed FDF dataset that suggests that pose information and a large model size is crucial for generating high-quality faces.

Our final model is trained for 18 days, 40M images, until we observe no qualitative differences between consecutive training iterations. It converges to a frèchect inception distance (FID) [7] of 1.51. Specific training details are given in Appendix A.

Anonymization method	Easy	Medium	Hard
No Anonymization	96.6%	95.7%	90.4%
Blacked out	20.6%	29.4%	44.2%
Pixelation (16x16)	94.2%	94.2%	**89.9%**
Pixelation (8x8)	91.0%	91.8%	88.6%
9x9 Gaussian Blur ($\sigma = 3$)	95.1%	92.2%	82.4%
Heavy Blur (filter size = 30% face width)	82.8%	85.5%	85.5%
DeepPrivacy (Ours)	**95.4%**	**94.4%**	89.6%

Table 1: **Face detection** average precision on the WIDER Face [27] validation dataset. The face detection method used is DSFD [14], the current state-of-the-art on WIDER-Face.

5.1 Effect of anonymization for face detection

To evaluate the impact of anonymization, we anonymize the WIDER-Face [27] validation set. We evaluate the average precision of a face detection method on the anonymized dataset and compare the results to the original dataset. Table 1 shows the average precision of different anonymization techniques. In comparison to the original dataset, DeepPrivacy only degrades the average precision by 1.2%, 0.9%, and 0.26% on the easy, medium and hard difficulties, respectively.

We compare DeepPrivacy anonymization to simpler anonymization methods; black-out, pixelation, and blurring. Figure 4 illustrates the different anonymization methods. DeepPrivacy generally achieves a significant higher AP compared to all other methods, with the exception of 16x16 pixelation. 8x8 pixelation suffers significantly in terms of average precision. We want to repeat that pixelation and blurring are shown to be insufficient anonymization methods, unable to remove all privacy-sensitive information [4, 19, 20].

Experiment details: For the face detector we use the current state-of-the-art, Dual Shot Face Detector (DSFD) [14]. The WIDER-Face dataset has no facial landmark annotation; therefore, we use the same method as we used for the FDF dataset to retrieve landmarks. To match the landmarks with a bounding box, we use the same greedy approach as earlier. Mask R-CNN [6] is not able to detect keypoints for all faces, especially in cases with high occlusion, low resolution, or faces turned away from the camera. Thus, we are only able to anonymize 43% of the bounding boxes in the validation set. Of the faces that are not anonymized, 22% are partially occluded, and 30% are heavily occluded. For the remaining non-anonymized faces, 70% has a resolution smaller than 14x14. Note, for each experiment in Table 1; we anonymize the same bounding boxes.

5.2 Ablation Experiments

We perform several ablation experiments to evaluate the model architecture choices. We report the average Frèchet Inception Distance [7] between the original image and the anonymized image for each experiment. We calculate FID from a validation set of $27,000$ faces from the FDF dataset. The results are shown in Table 2 and discussed in detail next.

Effect of pose information: Face poses provided as conditional information improves our model significantly, as seen in Table 2a. The FDF dataset has a large variance of faces in all different poses and we find it necessary to include sparse pose information to generate realistic faces. In contrast, when trained on the CelebA dataset, our model completely ignores the given pose information.

Model	FID
With Pose	**2.80**
Without Pose	3.01

(a) Result of using conditional pose.

Discriminator Architecture	FID
Deep Discriminator	5.04
Wide Discriminator	**2.80**

(b) Result of the deep and wide discriminator.

#parameters	FID
12M	2.80
46M	**2.01**

(c) Result of different model size.

Table 2: **Ablation experiments** on our architecture. We report the frèchet inception distance after showing the discriminator 20M images (lower is better). For results in Table 2a and Table 2b, we use a model size of 12M parameters for both the generator and discriminator.

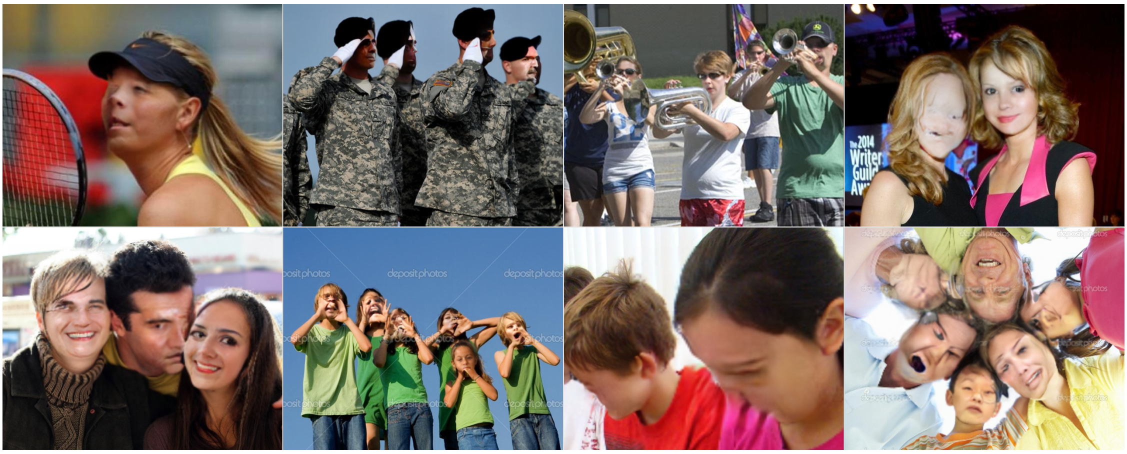


Figure 5: **Failure cases of DeepPrivacy** Our proposed solution can generate unrealistic images in cases of high occlusion, difficult background information, and irregular poses.

Discriminator Architecture: Table 2b compares the quality of images for a deep and wide discriminator. With a deeper network, the discriminator struggles to converge, leading to poor results. We use no normalization layers in neither of these networks, causing deeper networks to suffer from exploding forward passes and vanishing gradients. Even though, Brock *et al*. [2] also observe that a deeper network architecture degrades the overall image quality. Note, we also experimented with a discriminator with no modifications to number of parameters, but this was not able to generate realistic faces.

Model size: We empirically observe that increasing the number of filters in each convolution improves image quality drastically. As seen in Table 2c, we train two models with 12M and 46M parameters. Unquestionably, increasing the number of parameters generally improves the image quality. For both experiments, we use the same hyperparameters; the only thing changed is the number of filters in each convolution.

We further experimented with a model with 160M parameters. We had to reduce the batch size significantly, and we were unable to converge it to an image size of 128x128, due to computational limitations. The preliminary results on 64x64 resolution were slightly worse than our model with 46M parameters, probably due to the reduced batch size.

6 Limitations

Our method proves its ability to generate objectively good images for a diversity of backgrounds and poses. However, it still struggles in several challenging scenarios. Figure 5 illustrates some of these. These issues can impact the generated image quality, but our solution ensures the removal of all privacy-sensitive information.

Faces occluded with high fidelity objects are extremely challenging when generating a realistic face. For example, in Figure 5 several images have persons covering their face with hands. To generate a face in this scenario requires complex semantic reasoning, which is still a difficult challenge for generative adversarial networks.

Handling non-traditional poses can cause our model to generate corrupted faces. We use a sparse pose estimation to describe the face pose, but there is no limitation in our architecture to include a dense pose estimation. A denser pose estimation would, most likely, improve the performance of our model in cases of irregular poses. However, this would set restrictions on the pose estimator and restrict the practical use case of our method.

7 Conclusion

We propose a conditional generative adversarial network, *DeepPrivacy*, to anonymize faces in images without destroying the original data distribution. The presented results on the WIDER-Face dataset reflects our model's capability to generate high-quality images. Also, the diversity of images in the WIDER-Face dataset shows the practical applicability of our model. The current state-of-the-art face detection method can achieve 98.8% of the original average precision on the anonymized WIDER-Face validation set. In comparison to previous solutions, this is a significant improvement to both the generated image quality, and the certainty of anonymization. Furthermore, the presented ablation experiments on the FDF dataset suggests that a larger model size and inclusion of sparse pose information is necessary to generate high-quality images.

DeepPrivacy is a conceptually simple generative adversarial network, easily extendable for further improvements. Handling irregular poses, difficult occlusions, complex backgrounds, and temporal consistency in videos is still a subject for further work. We believe our contribution will be an inspiration for further work into ensuring privacy in visual data.

Appendix A - Training details

We use the same hyperparameters as Karras *et al*. [12], except the following: We use a batch size of 256, 256, 128, 72 and 48 for resolution 8,16,32,64, and 128. We use a learning rate of 0.00175 with the Adam optimizer. For each expansion of the network, we have a transition and stabilization phase of 1.2M images each. We use an exponential running average for the weights of the generator with decay 0.999, as this generally improves overall image quality[28]. Our final model was trained for 18 days on two NVIDIA V100-32GB GPUs.

Tensor Core Modifications

To utilize tensor cores in NVIDIA's new Volta architecture, we do several modifications to our network, following the requirements of tensor cores. First, we ensure that each convolutional block use number of filters that are divisible by 8. Secondly, we make certain that the batch size for each GPU is divisible by 8. Further, we use automatic mixed precision for pytorch[21] to significantly improve our training time. We see an improvement of 220% in terms of training speed with mixed precision training.

References

[1] Michael Boyle, Christopher Edwards, and Saul Greenberg. The effects of filtered video on awareness and privacy. In *Proceedings of the 2000 ACM conference on Computer supported cooperative work*, pages 1–10. ACM, 2000. ISBN 1581132220.

[2] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large Scale GAN Training for High Fidelity Natural Image Synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

[3] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[4] Ralph Gross, Latanya Sweeney, Fernando De la Torre, and Simon Baker. Model-based face de-identification. In *2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06)*, page 161. IEEE, 2006. ISBN 0769526462.

[5] Ralph Gross, Latanya Sweeney, Jeffrey Cohn, Fernando De la Torre, and Simon Baker. Face de-identification. In *Protecting privacy in video surveillance*, pages 129–146. Springer, 2009.

[6] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2980–2988. IEEE, 2017. ISBN 1538610329.

[7] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, pages 6626–6637, 2017.

[8] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1501–1510, 2017.

[9] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[10] Youngjoo Jo and Jongyoul Park. SC-FEGAN: Face Editing Generative Adversarial Network with User's Sketch and Color. *arXiv preprint arXiv:1902.06838*, 2019.

[11] Amin Jourabloo, Xi Yin, and Xiaoming Liu. Attribute preserved face de-identification. *Proceedings of 2015 International Conference on Biometrics, ICB 2015*, pages 278–285, 2015. doi: 10.1109/ICB.2015.7139096.

[12] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018.

[13] Tero Karras, Samuli Laine, and Timo Aila. A Style-Based Generator Architecture for Generative Adversarial Networks. *arXiv preprint arXiv:1812.04948*, 2018.

[14] Jian Li, Yabiao Wang, Changan Wang, Ying Tai, Jianjun Qian, Jian Yang, Chengjie Wang, Jilin Li, and Feiyue Huang. Dsfd: dual shot face detector. *arXiv preprint arXiv:1810.10220*, 2018.

[15] Yijun Li, Sifei Liu, Jimei Yang, and Ming-Hsuan Yang. Generative face completion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3911–3919, 2017.

[16] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2117–2125, 2017.

[17] Guilin Liu, Fitsum A Reda, Kevin J Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 85–100, 2018.

[18] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[19] Carman Neustaedter, Saul Greenberg, and Michael Boyle. Blur filtration fails to preserve privacy for home-based video conferencing. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 13(1):1–36, 2006. ISSN 1073-0516.

[20] Elaine M Newton, Latanya Sweeney, and Bradley Malin. Preserving privacy by de-identifying face images. *IEEE transactions on Knowledge and Data Engineering*, 17 (2):232–243, 2005. ISSN 1041-4347.

[21] NVIDIA. A pytorch extension: Tools for easy mixed precision and distributed training in pytorch. https://github.com/NVIDIA/apex, 2019.

[22] Zhongzheng Ren, Yong Jae Lee, and Michael S Ryoo. Learning to anonymize faces for privacy preserving action detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 620–636, 2018.

[23] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[24] Manuel Ruder, Alexey Dosovitskiy, and Thomas Brox. Artistic style transfer for videos. In *German Conference on Pattern Recognition*, pages 26–36. Springer, 2016.

[25] Latanya Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, 2002. ISSN 0218-4885.

[26] Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. YFCC100M: The new data in multimedia research. *arXiv preprint arXiv:1503.01817*, 2015.

[27] Shuo Yang, Ping Luo, Chen Change Loy, and Xiaoou Tang. Wider face: A face detection benchmark. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[28] Yasin Yazıcı, Chuan-Sheng Foo, Stefan Winkler, Kim-Hui Yap, Georgios Piliouras, and Vijay Chandrasekhar. The unusual effectiveness of averaging in gan training. *arXiv preprint arXiv:1806.04498*, 2018.

[29] Raymond A Yeh, Chen Chen, Teck Yian Lim, Alexander G Schwing, Mark Hasegawa-Johnson, and Minh N Do. Semantic image inpainting with deep generative models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5485–5493, 2017.

[30] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris Metaxas. StackGAN: Text to Photo-Realistic Image Synthesis with Stacked Generative Adversarial Networks. *Proceedings of the IEEE International Conference on Computer Vision*, 2017-Octob:5908–5916, 2017. ISSN 15505499. doi: 10.1109/ICCV.2017.629.

[31] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-Attention Generative Adversarial Networks. *arXiv preprint arXiv:1805.08318*, 2018.

[32] Shifeng Zhang, Xiangyu Zhu, Zhen Lei, Hailin Shi, Xiaobo Wang, and Stan Z Li. S3fd: Single shot scale-invariant face detector. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 192–201, 2017.