Utrecht University

MSc Artificial Intelligence

MASTER THESIS

Semantic-Aware Person Inpainting Using Generative Adversarial Networks

Author:
Derk van den Doel *(1982672)*

Supervisor:
Dr. ir. Ronald Poppe

Daily supervisor:
Metehan Doyran MSc

External supervisor:
Laurens Samson MSc

Second examiner:
Prof. dr. Albert Salah

May 2, 2023

Abstract

The widespread use of street view imagery in various applications, such as urban planning, navigation, and real estate, has raised concerns about the privacy of individuals captured in these images. While anonymization methods, such as blurring or pixelation, are commonly used to address these concerns, they often result in a loss of important information and can be easily circumvented by determined individuals. This thesis aims to explore an alternative approach to protecting privacy in street view imagery by using image inpainting techniques. Image inpainting involves filling in missing or obscured regions of an image in a way that preserves the overall structure and context of the scene. The proposed approach will utilize conditional Generative Adversarial Networks (cGANs) for the purpose of image inpainting, specifically in the context of street view images where people have been removed. While current cGAN-based methods rely solely on incomplete images with missing areas, our approach incorporates an additional source of information in the form of a semantic segmentation map. The semantic segmentation map serves as a prior to guide the inpainting process, allowing the model to better understand the context and structure of the image. By utilizing both the incomplete image and the semantic segmentation map, we aim to improve the quality and realism of the inpainted results. We compared the performance of our two proposed models, SemGAN and SemGAN-GT, with the existing pix2pix method. SemGAN utilizes inpainted semantic segmentation maps as a prior in image inpainting, while SemGAN-GT uses ground truth semantic segmentation maps. We evaluated the performance of each method using established image quality metrics, such as L1, SSIM, and PSNR, as well as a qualitative analysis of the inpainted images. Our findings indicate that both SemGAN and SemGAN-GT outperform the pix2pix method in terms of image quality and realism. This suggests that semantic information improves the quality and realism of the inpainted results, based on our quantitative results and qualitative analysis.

Contents

Chapter 1

Introduction

Since its release in 2007 Google Street View has collected and hosted more than 220 billion images from over 100 countries. Other smaller services later followed, such as Baidu Maps, Bing Maps and the crowdsourced service Mapillary. In this study "street view imagery" refers to all street-level images. The amount and density of the imagery offers users the possibility to wander through a virtual world, hereby enabling the user to search for real estate, virtual tourism, travel planning, enhanced driving directions, and business search (Frome et al., 2009). Access to such imagery also offers possibilities to companies and governmental organisations. Computer vision algorithms can be applied to the imagery, for instance, in city maintenance to localize garbage or detect broken assets.

While there is no doubt about the usefulness of street view imagery, it raises privacy concerns as the street-level images contain many personally identifiable features, such as faces and license plates (Flores and Belongie, 2010). These privacy concerns are not only limited to services employing street view imagery, it is part of a growing concern regarding data privacy and the misuse of technology (van Zoonen, 2016). Driven by these concerns, the anonymization of people in imagery has been a fast growing field of research in recent years, and will also be the topic of this study.

The motivation of this research is a joint project between the municipalities of Amsterdam and Utrecht. To optimize the service in cities, the municipalities want to implement a computer vision-based solution that, for example, is able to localize garbage or recognize broken assets. To accomplish this, a database with images from the city of Amsterdam has been constructed. However, the majority of these images contain persons, which raises privacy concerns and puts restrictions on the storage, processing and sharing of the data, as described in the GDPR. Moreover, the protection of privacy is the highest priority for governmental organisations. Therefore an initial solution to anonymize people was created, which involved the blurring of faces. However, this was deemed unsatisfactory as it was still possible to recognize people based on soft biometric and non-biometric identifiers. Hence, the municipalities seek for an alternative solution to anonymize people, protecting the privacy of individuals while still guaranteeing intelligibility.

The most conventional method to hide the identity of a person is to apply blurring to a region of interest, such as the face. This anonymization method has also been utilized in Google Street View (Frome et al., 2009). However, even after the application of this method, it is still possible to uncover a person's identity based on their clothing, hairstyle and geo-location. Moreover, recent research by McPherson et al. (2016) has shown that methods to recover identity from blurred images are becoming increasingly successful.

Although privacy protection is the main concern, it is also important that the information within the image is retained. A loss of information would inhibit the municipality to analyse important information, such as the recognition of the aforementioned broken assets or garbage. Therefore, the scene should both look realistic and the details must be preserved. The balance between the protection of privacy of individuals in an image and the retention of information in an image is called the privacy intelligibility trade-off (Hudson and Smith, 1996). Image filters such as blurring lack the ability to provide a balance between privacy protection and image utility, which generally happens at the expense of privacy loss

(Padilla-López et al., 2015).

Although image filtering methods have been the conventional method, it has become clear that these anonymization methods can be improved. Other more recent fields of research that could possibly offer an improvement are image generation and image inpainting. These fields both highly benefit from the introduction of Generative Adversarial Networks (GANs) (Goodfellow et al., 2014), and their generative capacities. The first solution involves the substitution of a human individual with a "fake" person generated by the GAN. While solutions exist that produce realistic results (Ouyang et al., 2018), the variation in samples is limited, which makes the images appear less realistic. The second solution involves the removal of human individuals from an image, followed by utilizing Generative models such as GAN to fill the missing region seamlessly. This technique is known as image inpainting. To summarise, the former method substitutes the real person with a generated fake person, while the latter method removes all people from the image and uses image inpainting techniques to complete the image.

GANs are renowned for textural synthesis and excel at background completion or removing objects, which are tasks that require repetitive structural synthesis (Lugmayr et al., 2022). Hence, inpainting has the potential to circumvent the privacy intelligibility trade-off, since the removal of people protects their privacy and the inpainting process retains the intelligibility.

1.1 Scope

For an anonymization method to be successful, it should remove all the personal data in an image, for all images in the database. In order to do so we must first establish a common understanding about the meaning of 'personal data'. In the European General Data Protection Regulation (GDPR) personal data is described as "any information relating to an identified or identifiable natural person ('data subject'); an identifiable natural person is one who can be identified, directly or indirectly, in particular by reference to an identifier such as a name, an identification number, location data, an online identifier or to one or more factors specific to the physical, physiological, genetic, mental, economic, cultural or social identity of that natural person" (*European General Data Protection Regulation* 2018). Most of the identifiers in this definition will not be present in imagery. Therefore, we will only focus on the factors specific to the physical identity of a person. Moreover, our method does not concern vehicles and therefore will not be able to remove or inpaint license plates, or cars and other vehicles. We will train and test our model using daytime street-level imagery of various European cities.

1.2 Contributions

We propose a GAN-based inpainting method capable of anonymizing street-view images while retaining the intelligibility of the images.

1.3 Research questions

Based on the given problem description and the proposed solution, we formulate the following research question:

> *Can a Generative Adversarial Network anonymize street level images, in which human individuals are removed, while the intelligibility of the images is retained?*

In order to answer the main research question, we divide it into the following sub-questions:

> **SQ1:** *How does the inclusion of semantic segmentation data influence the quality of the generated images?*

The meaning of quality in SQ1 is twofold. Anonymization is the main goal, therefore quality concerns the ability to protect the privacy of human individuals in an image. In addition, the overall realistic appearance of an image, as well as the preservation of detail are important and therefore also apply to quality.

The performance of our inpainting model will be evaluated on a well-known dataset that serves as a benchmark for image inpainting algorithms. This dataset contains daytime street-level images similar to the dataset of Amsterdam belonging to the municipality. In addition, the dataset provides pixel-level annotations for each image, which means that each pixel is labeled with its corresponding object class. This semantic segmentation map provides a rough outline of the objects and regions present in the image, which can guide the GAN in filling in missing regions of the image in a way that is consistent with the surrounding context.

We will assess the effectiveness of our inpainting model by utilizing commonly used metrics in research, namely the L1 distance and peak signal-to-noise ratio (PSNR). Additionally, we will incorporate the structural similarity index (SSIM) as a means of ensuring that the generated results are not only quantitatively accurate but also perceptually accurate. To calculate these metrics, we will compare the ground truth images with their respective inpainted images. This approach is based on previous research findings that demonstrate the potential for L1 and PSNR scores to inaccurately reflect perceptual accuracy (Nazeri et al., 2019). To evaluate the effectiveness of our inpainting model, we will compare its performance to an existing model based on the pix2pix architecture (Isola et al., 2017).

By incorporating semantic segmentation maps as a prior, we expect a refinement in the overall visual realism of the inpainted image, as evaluated through visual inspection, as well as an improvement in the metrics mentioned above. The effectiveness of the addition of a prior in GAN-based inpainting has been shown by Nazeri et al. (2019). Therefore we anticipate that the inclusion of additional information in the form of semantic segmentation data will produce improved results.

SQ2: *How does the reduction of training data influence the performance of the model?*

A common method to improve the performance of deep learning models is to increase the number of training samples used. As we already use the entire finely annotated Cityscapes dataset this is not possible. The pattern we expect to see when increasing the training dataset would be that the models performance improves, but with diminishing returns. We can however hypothesize about the effect of enlarging the training dataset by evaluating a model's performance as we train it on increasingly larger proportions of our training dataset. We expect to see a similar pattern where we the model performs well when using the entire dataset or a substantial portion of it, but experiences a notable decrease in performance when the training samples are reduced.

1.4 Thesis Outline

The remaining sections of this work are structured as follows: Chapter 2 summarizes related work concerning visual personal identifiers, detection of sensitive information, image anonymization, generative models and metrics used to evaluate these models. Chapter 3 describes our methods for generating realistic anonymized images, and the evaluation of these images. In Chapter 4 of this thesis, we will present both quantitative and qualitative results and subsequently discuss them. To conclude this thesis, we provide our final conclusions and outlook on future work in Section 5.

Chapter 2

Literature review

In this chapter we will give an overview of the most relevant and significant publications related to our problem. In Section 2.1 we discuss visual identifiers, separated in biometric, soft biometric and non-biometric identifiers. Section 2.2 introduces various privacy protection methods used within the field. Section 2.3 covers Generative Adversarial Networks and their application to image inpainting problems. Section 2.4 describes metrics used to assess the performance of anonymization techniques.

2.1 Visual Personal Identifiers

When gathering image data in the public domain it is inevitable that information about human individuals is captured in these images. Besides the ethical responsibilities we have concerning privacy protection, restrictions with respect to data collection, processing and sharing imposed in the European General Data Protection Regulation (GDPR) serve as a manual in handling personal data. If our algorithm is able to delete all personal data, individuals are no longer identifiable. Therefore the principles of data protection do not apply anymore to the anonymized data, as described in recital 26 of the GDPR (*European General Data Protection Regulation* 2018). Hence, this allows for the data to be collected, stored and processed legally.

The physical and physiological identifiers can be classified according to the taxonomy presented in the work of Ribaric et al. (2016), in which they define three categories: biometric, soft-biometric and non-biometric identifiers.

2.1.1 Biometric identifiers

Biometric identifiers can be categorized as either physiological such as face, fingerprint and iris; and behavioural including voice, gesture and gait. These identifiers are the distinctive, measurable, generally unique and permanent personal characteristics used to identify individuals (Ribaric et al., 2016). Regarding the behavioural identifiers, sound is absent in image data and therefore can be ignored. Gait can be described as the combination of stride and walking speed, this could be extracted from video recordings along with gesture and can be used to identify a person (Jia et al., 2017). However, our data consists of panorama images taken regularly, after every x meters. Due to the missing information between two frames, it is impossible to extract useful information about gait and gestures of individuals. As for physiological identifiers, we assume that because of their small size, it is impossible to extract fingerprint or iris information.

The main identifier used in images to recognize human individuals is the face. Therefore, methods regarding the anonymization of faces have dominated research in recent years. In the early days of anonymization most research was done on "blurring" and "pixelating" images or parts of images (C. Zhang et al., 2006; Frome et al., 2009; Boyle et al., 2000). However, recent research has shown that it is still possible to extract useful information from images with blurred or pixelated faces (Brkic et al.,

2017). This suggests that only anonymizing the face is not enough to protect the privacy of a human individual.

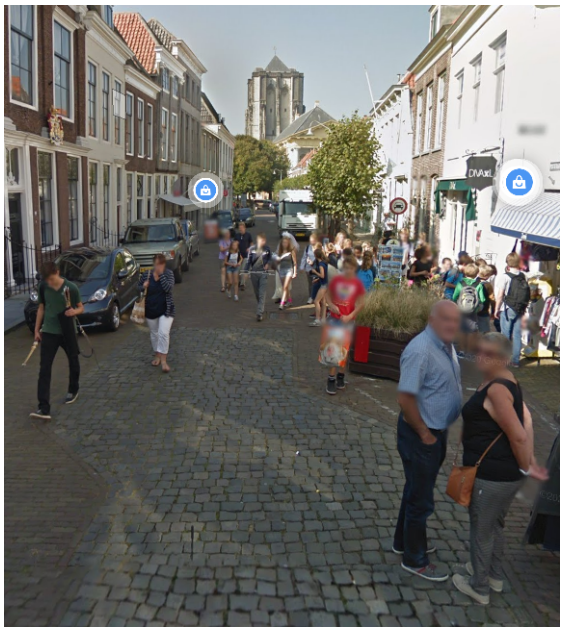


Figure 2.1: Example image from Google Street View. A blurring filter is applied on faces, leaving soft and non-biometric identifiers intact.

2.1.2 Soft biometric Identifiers

As opposed to biometric identifiers, soft biometric identifiers can be used to describe a person, this can be done by saying something about the appearance of an individual, such as their height, weight, hair color, ethnicity and indelible marks like scars, birthmarks or tattoos. These features are neither unique to an individual, nor permanent, they can change over time (growing, dyeing hair). Therefore, these identifiers are not sufficient to make a distinction between two individuals (Jain et al., 2004). However, soft biometric identifiers have a lot of potential when they are combined with biometric identifiers to improve the recognition performance (Reid et al., 2013).

Consider a Google Street View image where an individual is visible but their face is anonymized (see Figure 2.1). Regardless of the absence of biometric identifiers, based on location, soft biometric and non-biometric identifiers it could still be possible to identify an individual (Reid et al., 2013). It follows that soft biometric identifiers carry privacy-intrusive information about an individual and should therefore be removed.

We assume that when the skin and face are anonymized, it is impossible to extract information about indelible marks, race or ethnicity. In addition, we assume that height, weight and hair color are inadequate to identify an individual, if not combined with biometric data.

2.1.3 Non-biometric Identifiers

Non-biometric identifiers consist of all personal identifiers that are not components of the physical or behavioural features of an individual. The identifiers convey information about a person, but are both non-permanent and changeable. These mainly are: hairstyle, clothing and license plates (Ribaric et al., 2016). In our research we focus on the anonymization of persons, we assume that license plates in our data are already blurred and can therefore be ignored.

The importance of anonymizing the non-biometric identifiers is shown by Brkic et al. (2017). In their previous research they show that a re-identification algorithm is able to recognize people, even

Figure 2.2: Blurring applied to the full silhouette of individuals. Even after the anonymization process, soft and non-biometric identifiers such as hairstyle and clothing remain recognizable. Images from the Clothing Co-Parsing (CCP) dataset (W. Yang et al., 2014)

after the faces of these individuals are anonymized. Even when the entire silhouette of the individual is blurred, the re-identification algorithm's performance is still 40%. From this we can conclude that non-biometric identifiers are important features in the identification of individuals (see Figure 2.2), even if these identifiers are rarely addressed as a problem in research (Hrkać et al., 2017).

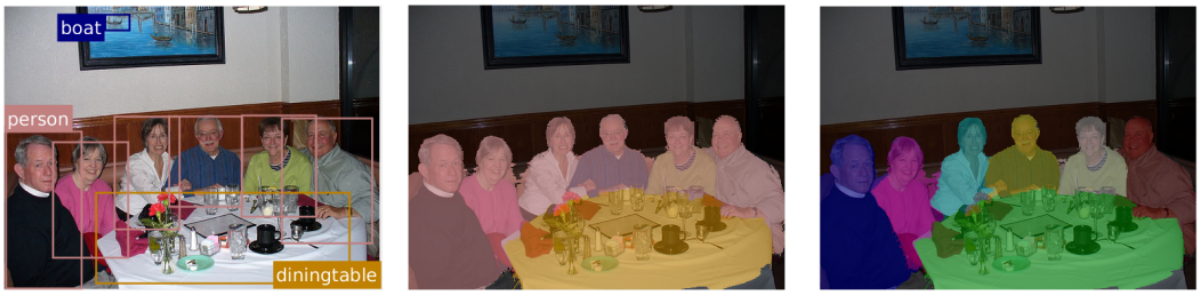
2.2 Detection of sensitive information

The process of detecting regions of interest (RoIs) is called object detection. In our project we regard sensitive areas as RoIs. A sensitive area is a region in an image in which visual personal identifiers are shown. Detecting these regions is crucial, as this is the first step in anonymizing the image.

Object detection is an important computer vision task with the objective to locate and classify instances of certain classes (such as faces, cars or animals) in an image. Typically, object detection algorithms locate objects by determining the coordinates of an object and drawing a bounding box around the objects of interest. When a more fine-grained method is required, we turn to semantic segmentation. This method allows for a more precise understanding of the image by assigning a class label to each pixel in the image. Semantic segmentation is also related to instance segmentation, which aims at assigning a unique label to each segmented object. For more details, see Figure 2.3.

It is widely accepted that two major historical periods have shaped the development of object detection: "traditional objected detection" (before 2014) and "deep learning based methods" (after 2014) (Zou et al., 2019).

Object detection gained huge momentum through deep learning, and more particularly the Convolutional Neural Network (CNN), introduced by Krizhevsky et al. (2012a). Before CNNs, object detection algorithms mostly relied on handcrafted feature representations. However, due to nuisance factors in images it is very difficult to design a feature descriptor that describes all types of objects. In comparison, CNNs can learn both low-level and high-level image features, these learned features are more representative than the handcrafted features (Youzi et al., 2020). In Section 2.2.1 we will discuss traditional object detection methods. Section 2.2.2 outlines the current state-of-the-art deep learning-based object detection methods



Object Detection | Semantic Segmentation | Instance Segmentation

Figure 2.3: Object detection needs to identify the category of objects and locate the objects with rectangular bounding boxes. Semantic segmentation predicts the categories of each pixel. Instance segmentation needs to predict both the categories of each pixel and object instances. Image from (Arnab et al., 2018)

2.2.1 Traditional object detection methods

A milestone in the pre-CNN era was the movement from global representation to local representation that were invariant to nuisance factors in images (L. Liu et al., 2018).

One of the first and most renowned algorithms that used these local representations is the Scale Invariant Feature Transform (SIFT) algorithm (Lowe, 2001). SIFT descriptors are scale and rotation invariant. This is achieved by calculating the gradient of certain keypoints at different scales of the image. The location, orientation and scale of each keypoint is stored in a histogram, which subsequently can be used in object detection by comparing the SIFT descriptors of two objects. The main disadvantage of SIFT however, is the speed of the algorithm. Inspired by the work of Lowe (2001), Bay et al. (2006) proposed an algorithm similar to SIFT called "Speeded up robust features" (SURF). Where SIFT uses Gaussian averaging to find keypoints, SURF applies convolution on the integral image, which is faster, especially since the calculations can be done in parallel for different scales (Karami et al., 2015). Comparative studies prove that SURF indeed is faster than SIFT (Karami et al., 2015; Mistry and Banerjee, 2017; Luo and Oubong, 2009), but also indicate that SIFT descriptors are more robust than their SURF counterparts.

Two decades ago, Viola and Jones achieved real-time detection of faces for the first time without any constraints (Viola and Jones, 2001). Their algorithm, which was later referred to as the "Viola-Jones(VJ) detector", incorporated three important techniques to attain its detection speed: "Integral image", "Feature selection" and "Detection cascades" (Zou et al., 2019). The integral image is an intermediate representation of the image that allows for fast computation of rectangular filters. More specifically, the speed of this algorithm comes from the fact that the computations are independent of the size of the filters. The algorithm extracts Haar-like features (Papageorgiou et al., 1998) from the image, which are selected using the Adaboost algorithm. Adaboost lowers the weight of features that tend to produce false negatives, thereby it minimizes the number of these false negatives. The detection cascades quickly discard non-faces and thereby reduce the computational overhead. Although this algorithm performs very well on faces and other simple objects characterized by a small number of salient features, the algorithm would perform worse on highly textured objects. Moreover, the training phase is slow and can easily end in a classifier that produces many false positives (Andreopoulos and Tsotsos, 2013).

The Histogram of Oriented Gradients (HOG) feature descriptor by Dalal and Triggs (2005) is by many researchers regarded as an important improvement on SIFT (Zou et al., 2019). Like the VJ-detector, HOGs use a sliding window approach. The HOG algorithm divides the image in cells and encodes gradient information of that cell in a histogram. This results in a HOG feature vector containing information about an object. By using the angle and magnitude, HOGs are quite robust to local variations, such as translation and scale. Block normalization can be applied to be more robust to changes in illumination. The HOG feature vector can be used as an input into a detection algorithm. The authors opted for an SVM-based window classifier. Although it was implemented as a pedestrian detector, HOG-detectors can be applied to a variety of object classes.

The deformable Part Model (DPM) by Felzenszwalb et al. (2008) was originally proposed as an

extension on the HOG detector. This method decomposes the problem of detecting an object into detecting the individual parts of an object. Hence, the problem shifts from detecting a person into detecting arms, legs and a head. Typically, a DPM consists of a root-filter and several part-filters. Instead of manually specifying the configurations, the authors use latent Support-Vector Machines (LSVMs) to learn the possible configurations of these parts directly from images. A hard negative mining technique is used to improve training of the LSVMs. The authors later extended this method by implementing mixture models to account for the variantions within object classes (Felzenszwalb et al., 2010). Although DPM was state-of-the-art at its time, Deep Learning has exceeded these models in performance. However, methods such as bounding box regression, hard negative mining and mixture models have had great effect on Deep Learning methods.

Traditional local feature-based approaches have been very successful in the field of Computer Vision. However, it is generally acknowledged that these local feature-based approaches offer insufficient complexity in the types of object representations. More complex object representations are required to bridge the semantic gap between low level and high level representations (Andreopoulos and Tsotsos, 2013)

2.2.2 Deep learning-based object detection

The introduction of Alexnet (Krizhevsky et al., 2012b) resulted in various methods that attempted to bridge the gap between image classification and object detection. When we speak about CNN-based object detection methods, we generally make a distinction between one-stage detection and two-stage detection, or region-based proposal methods, as called by R. Girshick et al. (2013).

As the name suggests, a two-stage detection method has two stages: the region proposal stage and the region classification stage. The first stage proposes category-independent regions, a CNN then extracts features and the proposal is classified. Popular two-stage methods are Region-based CNN (RCNN) (R. Girshick et al., 2013) and all its variations, Spatial Pyramid Pooling net (SPPnet) (K. He et al., 2014) and DetectoRS (Qiao et al., 2021).

One-stage detection methods have another approach in which the region detection and classification process is not separated. A one-stage detection model consists of two components: a backbone model and a Single-Shot Detector(SSD) head. The backbone model is usually a pre-trained CNN network for classification. The SSD-head consists of one or multiple convolutional layers of which the outputs represent the bounding boxes and classifications. The most famous one-stage detector is the You Only Look Once (YOLO) detector (Redmon et al., 2015) and all its improved versions. Other one-stage methods are: Feature Pyramid Network (FPN) based methods RetinaNet (T.-Y. Lin et al., 2017) and EfficientDet (Tan et al., 2019), and Swin Transformer (Z. Liu et al., 2021).

Generally speaking, we can make a distinction between one-stage and two-stage detectors in terms of accuracy and speed (see Figure 2.4). The two-stage detectors have higher detection accuracy, whereas one-stage detectors excel in processing frames per second (fps). Therefore, tasks that require real-time detection (>30 fps) benefit more from using one-shot detectors, as two-stage detectors do not allow for real-time detection.

In the next subsection we will describe the object detection methods mentioned previously, and how they build upon each other in order to improve performance.

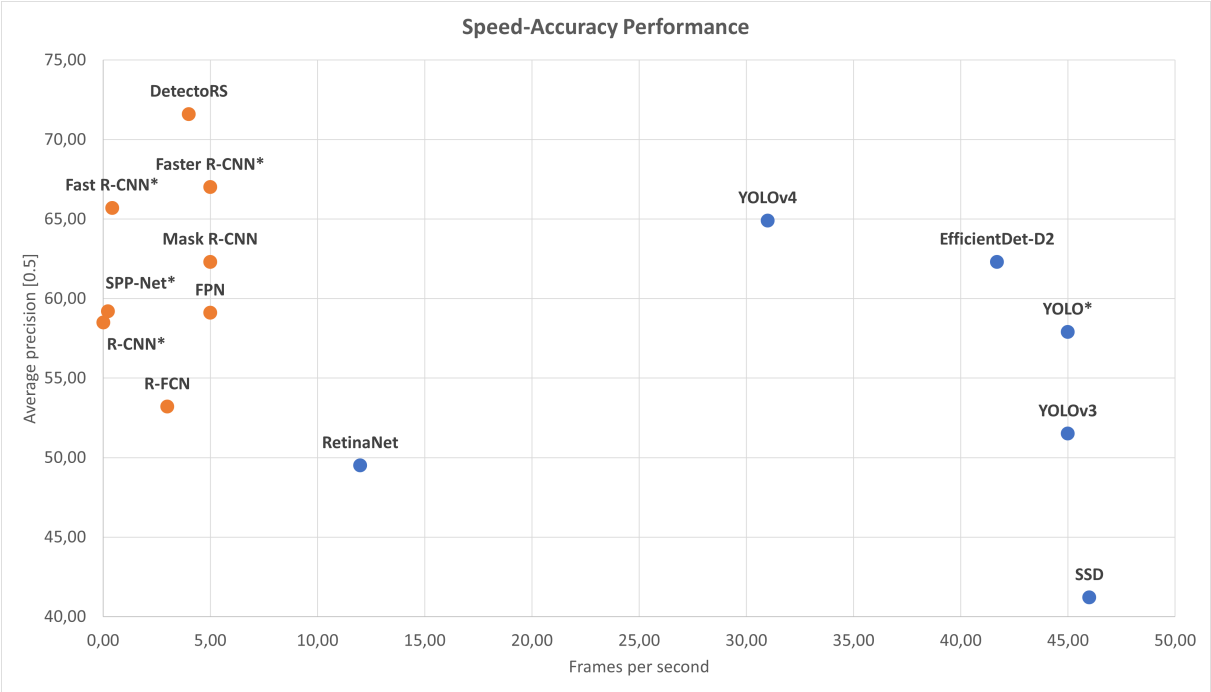


Figure 2.4: Speed-Accuracy performance of various object detectors. The orange points are two-stage detector models and the blue dots represent one-stage detectors. Models marked with a star are compared on PASCAL VOC 2012, while others on MS COCO. Data from (Zaidi et al., 2022)

Two-stage object detection methods

RCNN: RCNN was one of the first deep learning-based object detection methods. The method showed the potential of CNNs in object detection, as RCNN improved the best performing traditional object detection method by 30% on PASCAL VOC 2012. Figure 2.5 shows the flow of RCNN, the process can be divided in the following three steps:
Region proposal generation. Selective Search algorithm (Uijlings et al., 2013) is applied, which generates region proposals for each image. Selective Search is an iterative process in which groups of pixels are merged together based on their similarity in color, texture, shape and size. This is a bottom-up approach which generates region proposals from small, early on in the process, to larger, later in the process. This results in around 2000 region proposals which could potentially contain an object.
CNN-based feature extraction. Each region is warped (or cropped) to a fixed size resolution and fed into the CNN. The network outputs a feature vector as a final representation.
SVM classification. The feature vectors serve as input for class specific SVMs that predict the presence of an object and classify this object. To fine-tune the bounding boxes that do contain an object, bounding box regression is used to refine the bounding box, and Non-maximum suppression (NMS) is used to remove overlapping region proposals.

While RCNN greatly improved traditional object detection performance, there are some notable disadvantages. As it is a multi-stage pipeline, fine-tuning is cumbersome; the Selective Search algorithm is time-consuming, approximately 2 seconds to generate 2000 region proposals; features are extracted from each region proposal separately, feature sharing is therefore neglected as new features are calculated for each region proposal.
SPP-net: K. He et al. (2014) proposed SPP-net in an attempt to improve the computational speed of RCNN. This is accomplished with the addition of an Spatial Pyramid Pooling (SPP) layer, and a clever method which extracts image features only once.

Fully connected layers require a fixed-size input. RCNN solved this problem by cropping/warping each region proposal to a fixed resolution. However, this increases the computational load and leads to the loss of information in the input image. SPP-net removes the cropping/warping process and adds an SPP layer after the last convolutional layer. The SPP layer applies max pooling to the input in various

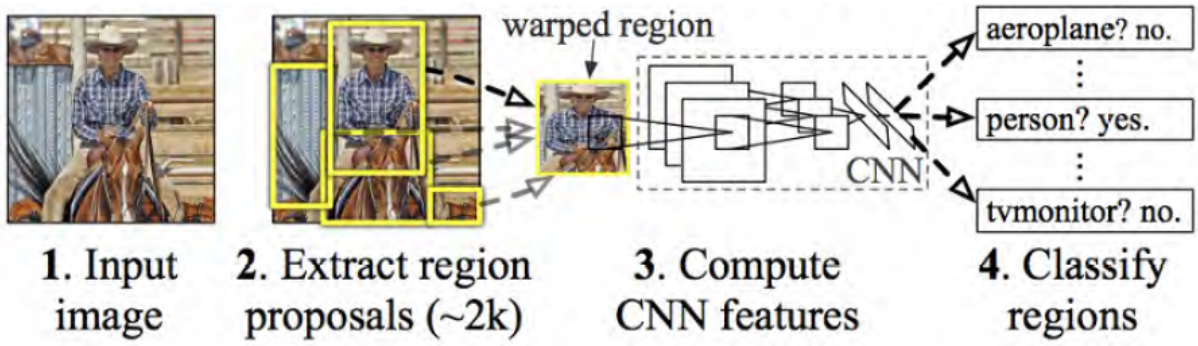


Figure 2.5: High-level overview of RCNN. It takes an image as input, applies a region proposal algorithm to generate candidate bounding boxes, a CNN is applied to each region to extract features, finally an SVM classifies the objects within the proposed regions. Image from (R. Girshick et al., 2013).

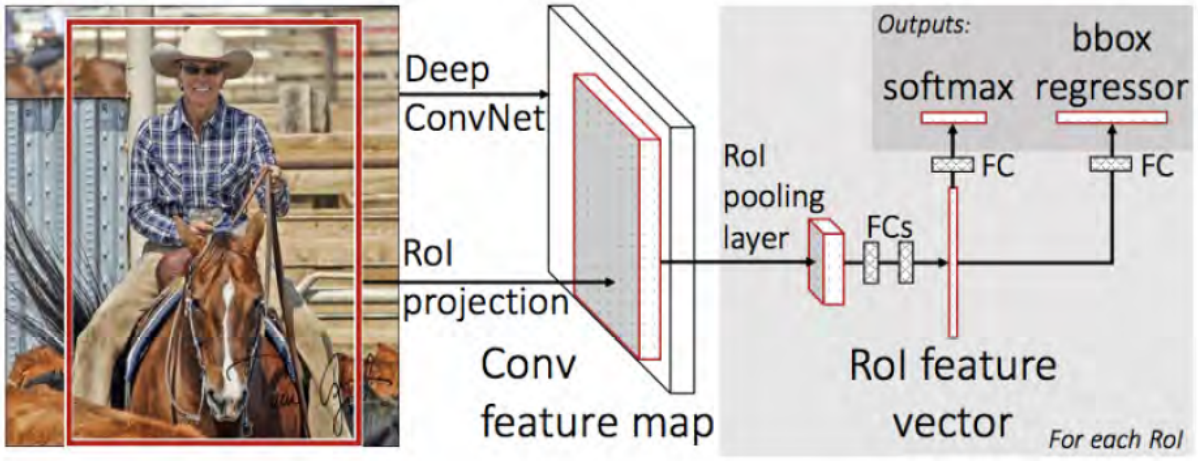


Figure 2.6: High-level overview of Fast RCNN. The region proposals are projected onto the feature map, RoI pooling is applied to construct a fixed-size feature vector, this vector is used as an input for the fully connected layers which output a softmax score and bounding box regression score. Image from (R. B. Girshick, 2015).

scales, in a pyramidal structure, and concatenates the outputs of these various pooling operations. As a result, the SPP layer always produces a fixed-size feature vector.

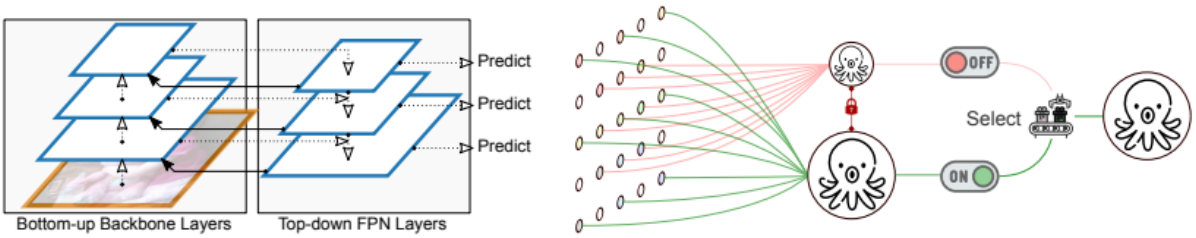
While SPP-net still uses Selective Search (like RCNN), the implementation is different. Each region proposal is projected on the feature map, which subsequently are converted into fixed-size feature vectors by the SPP layer. This therefore avoids repeated computations by the CNN, for each region proposal.

Despite the increase in detection speed over RCNN, with comparable accuracy, SPP-net still has some drawbacks: The pipeline is still multistage, and the SPP layer inhibits fine-tuning of the CNN layers during training.

Fast RCNN: R. B. Girshick (2015) proposed Fast RCNN, that builds upon RCNN and SPP-net in order to improve the detection speed and accuracy. In Fast RCNN, the pyramidal structure of SPP-net is replaced by an RoI (Region of Interest) pooling layer, which applies pooling to only one pyramid level to extract a fixed-size feature vector (see Figure 2.6). Each feature vector is used as an input to the fully connected layers, and then branches out to a softmax layer and a bounding box regression layer. The multi-stage training problem is solved by introducing multi-task loss and the softmax layer. The former allows for the training of both the softmax layer and the bounding box regression layer, the latter moves away from one-vs-rest linear SVMs that have to be trained separately.

While Fast RCNN was proposed as an improvement on the speed of RCNN, the introduction of end-to-end learning also led to an increase in detection accuracy.

Faster RCNN: The bottleneck in Fast RCNN is the region proposal algorithm, Selective Search takes



(a) Macro Design: Recursive Feature Pyramid.

(b) Micro Design: Switchable Atrous Convolution.

Figure 2.7: (a) shows the Recursive Feature Pyramid architecture, the solid arrows show the feedback connections from the top-down to the bottom-up pathway, this allows the network to look at the image twice or more. (b) shows the Switchable Atrous Convolution, this operation allows the network to look at the image with different atrous rates and combines this information using switches. Image from (Qiao et al., 2021).

2 second per image, while EdgeBoxes takes 0.2 seconds, which is equal to the time consumed for the detection. It is apparent that traditional region proposal algorithms lack the speed of Deep Learning-based algorithms. Therefore, S. Ren et al. (2015) propose Faster RCNN, which applies a Region Proposal Network (RPN) that shares features with the CNN.

Similar to Fast RCNN, the convolutional part of Faster RCNN, extracts features from the input image once. The region proposals are generated by RPN that uses a sliding window approach on the feature map. At each sliding window location, the RPN predicts multiple region proposals with varying scales and aspect ratios. Finally, for each region proposal,the RPN outputs 4 bounding box coordinates and a score that estimates the presence of an object in that region. The coordinates and confidence scores are refined by training the RPN separately from the Fast RCNN pipeline.
Mask RCNN: K. He et al. (2017) proposed Mask RCNN, a more fine-grained approach that applies instance segmentation. Mask RCNN is an extension of Faster RCNN that adds an extra branch for pixel-level object segmentation, this branch is a fully convolutional network (FCN).

Using the RoI pooling layer from Faster RCNN leads to the problem of misalignment between the RoI and the extracted features, which in turn leads to inaccuracies in segmentation. This problem arises because RoI pooling uses quantization to divide the feature maps in bins. Therefore, the authors introduce an ROIAlign layer, which uses bilinear interpolation instead of quantization to compute the bins.
Feature Pyramid Network (FPN): Most deep learning-based object detection methods only utilize the information from the last layer of a CNN for detection and classification of an object. However, T.-Y. Lin et al. (2016) applied the idea of image pyramids to deep learning, where they use intermediate feature maps from the CNN as an hierarchical feature pyramid, this is the bottom-up pathway. The top-down pathway is used to upsample the feature maps. The two pathways are laterally connected using 1x1 convolutional filters. Combined with Fast RCNN, FPN enhances detection accuracy, as the feature pyramid can extract semantics from all levels. Since its introduction, FPN has become a standard building block for later object detectors.
DetectoRS: The previous mentioned methods all apply the idea of looking at an image twice, once for the region proposal and once for feature extraction. Qiao et al. (2021) propose DetectoRS, in which they extend this idea to the backbone network. At macro-level, the authors propose Recursive Feature Pyramid (RFP), which incorporates feedback connections between the top-down and bottom-up pathways in FPN (see Figure 2.7a). The recursive nature of the model allows for the generation of increasingly powerful representations. At micro-level, Switchable Atrous Convolution (SAC) is introduced, which replaces regular 3x3 convolutions in the backbone network (see Figure 2.7b). SACs are able to apply convolution at various atrous rates, determined by the switch function, this is an average pooling layer combined with 1x1 convolution.

DetectoRS reaches state-of-the-art accuracy for two stage networks, yet with a processing speed of 4 fps, it is unsuitable for real time detection.

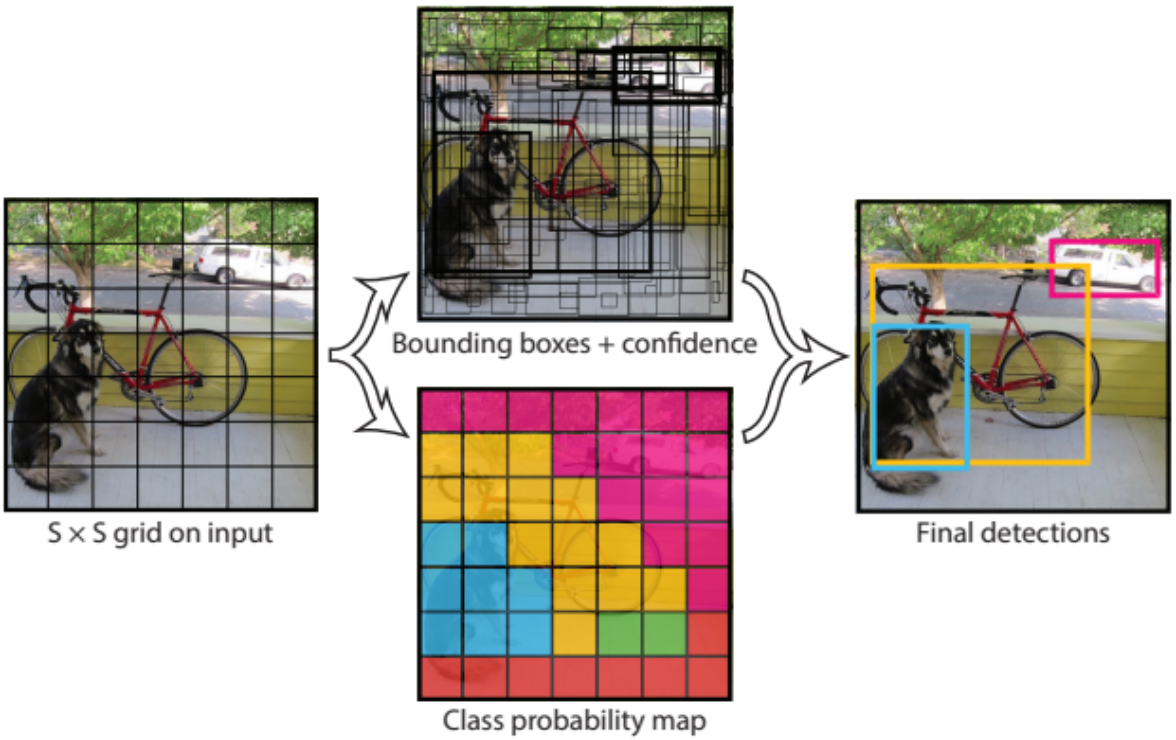


Figure 2.8: High level overview of YOLO. Image from (Redmon et al., 2015).

One-stage object detection methods

YOLO: The division between object detection and classification in two-stage detection methods improve the detection accuracy, but also leads to slower detection speed. Redmon et al. (2015), however, framed object detection as a regression problem turning image pixels to bounding box coordinates and class probabilities. By unifying the separate components of object detection, and evading region proposal methods, which are computationally more expensive, the detection speed is dramatically increased. While methods such as Faster RCNN are able to process 5 fps, YOLO can process 45 fps with base network, and up to 155 fps using a faster version.

YOLO divides the input image in a SxS grid, each cell in the grid is responsible for predicting the object centered in that cell. Furthermore, each cell predicts class probabilities, bounding box coordinates and confidence scores for these boxes. Bounding boxes with low confidence scores are discarded and overlapping bounding boxes are removed using non-maximum suppression (see Figure 2.8).

Although YOLO is able to perform real-time detection, it has several disadvantages. Each grid cell can predict only two bounding boxes and one object class, this limits the number of nearby objects that can be detected. Besides, by learning bounding boxes from data, the ability to generalize objects in new configurations is limited.
YOLOV2: With the objective of improving the accuracy of YOLO, Redmon and Farhadi (2017) propose YOLOV2. The first enhancement the authors introduce, is the addition of batch normalization which improves convergence and makes other regularization methods redundant. Second, the CNN is fine-tuned on higher resolution images to increase the detection of the various object classes. Third, multi-scale training allows for better detection of small objects. Fourth, instead of predicting the bounding boxes directly, anchor boxes are used, similar to the RPN in Faster RCNN. However, where faster RCNN uses anchor boxes of predefined size, YOLOV2 learns the size of the anchor boxes using kmeans clustering.
RetinaNet: Lin et al. T.-Y. Lin et al. (2017) claim that the accuracy gap between one-stage and two-stage detection methods is caused by the "foreground-background imbalance". Therefore, they propose RetinaNet in which their main contribution is the introduction of focal loss. This reshaped loss function focuses more on hard negatives by reducing the influence easy examples have on the loss. RetinaNet

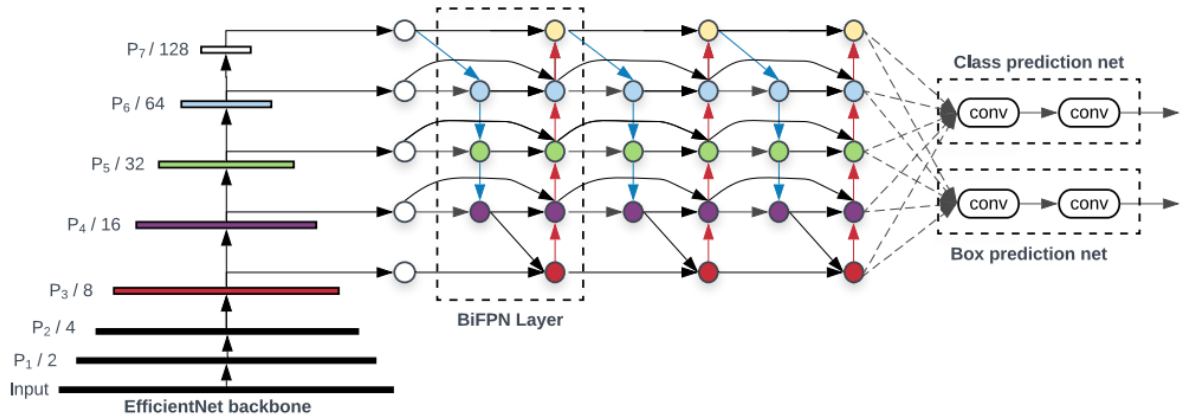


Figure 2.9: High level overview of EfficientDet. As a backbone network it recruits EffcientNet, BiFPN as feature network and shared class and box as its prediction network. As the backbone network is scaled up, the number of BiFPN and box/class layers will be increased accordingly. Image from (Tan et al., 2019).

uses FPN to improve multiscale detection. Each level of the FPN is processed by two subnets. The classification subnet predicts the probability of object presence at each location, the box regression subnet regresses the offset for anchor to the ground truth. These subnets are both FCNs that share parameters across networks.

YOLOV3: In YOLOV3 (Redmon and Farhadi, 2018) the authors present "incremental improvements" for the previous YOLO versions. The model makes use of a different backbone model called Darknet-53, a deeper CNN which makes use of residual layers. Furthermore, the model uses a method similar to FPN, which extracts features at three different scales. The small scale feature maps contain more fine-grained information, and the upsampled feature maps contain richer semantic information.

Although YOLOV3 presented some improvements, and performed better than earlier YOLO versions, it lacked ground breaking changes. Even RetinaNet had better accuracy than YOLOV3, while it came out a year earlier.

EfficientDet: In most object detection methods, we see a trade-off between accuracy and speed, where most of the detectors prove to be only good in one. With EfficientDet Tan et al. (2019), propose a new family of object detectors aiming to improve efficiency. The paper offers three major contributions. BiFPN, a bidirectional feature network. Compound scaling, a new method to jointly scales the resolution and size of the network. And EfficientDet, a new family of detectors with better accuracy and efficiency.

The EfficientNet extracts multi-scale features that are fused by the BiFPN layer (see Figure 2.9). A regular FPN is limited to a one-way information flow, as the name suggests, BiFPN allows information to flow efficiently in both ways. Most previous networks simply sum the input features in the fusing process, however, input features at different resolutions contribute to the output unequally. Therefore, BiFPN assigns learnable weights to learn the importance of each input feature. By adding more consecutive BiFPN layers, more high-level feature fusion is enabled.

The authors propose a compound scaling method, which uses a compound coefficient to jointly scale-up all dimensions of the EfficientNet, the number of BiFPN layers, box/class prediction layers and the resolution of the input images.

EfficientDet is easily scalable, achieves better accuracy and efficiency than previous detectors, while being smaller and computationally cheaper. The authors also show that the model can be used for semantic segmentation, by only using P_2 for the final per-pixel classification.

YOLOV4: In YOLOV4 (Bochkovskiy et al., 2020) the authors present a number of improvements to enhance both speed and accuracy as well as training time. Most existing deep learning-based object detectors require multiple GPUs for training, whereas YOLOV4 can be trained on a single GPU.

The authors make a distinction between two different groups of methods that can increase the accuracy: Bag-of-Freebies, methods that only increase the training time; and Bag-of-Specials, methods that slightly increase the inference time.

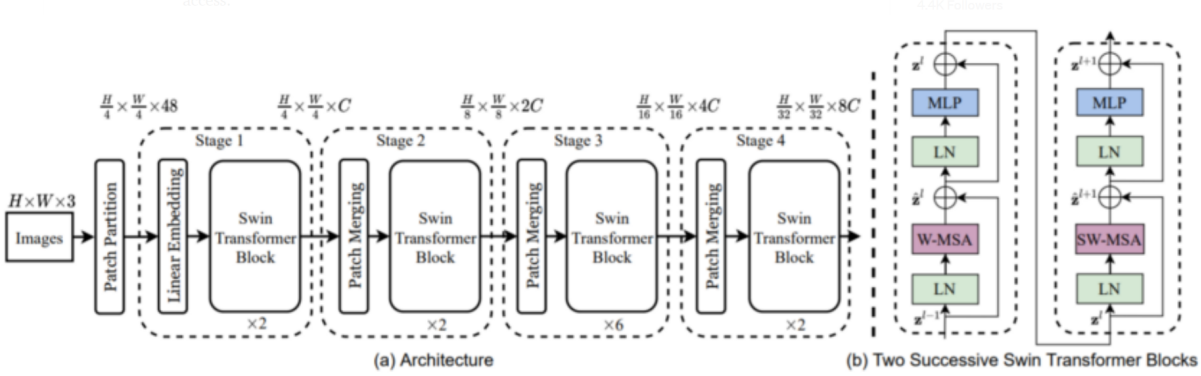


Figure 2.10: (a) High level overview of the Swin Transformer. (b) two successive Swin Transformer Blocks, where W-MSA and SW-MSA are multi-headed self attention modules with regular and shifted windowing configurations. Image from (Z. Liu et al., 2021).

Many Bag-of-Freebies methods are applied, but here we will only discuss the most important. The authors introduce Mosaic, a data augmentation method which is an improvement on CutMix. Mosaic mixes 4 training images, which allows for detection outside normal context of the objects. Another data augmentation method introduced is Self Adversarial Training (SAT). SAT works with 2 forward-backward passes of the network. In the first backward pass the weights are not updated, but some amount of perturbation is added to the image. In the second forward pass the perturbed image is used to train the network. SAT allows the model to reduce overfitting and generalize better. The authors also introduce Cross-Iteration mini Batch Normalization (CmBN), a modification on Cross-iteration Batch Normalization (CBN). CBN is a method that uses the statistics of the last 4 batches to update the current batch. CmBN only uses the statistics between mini-batches within a single batch.

The most important Bag-of-Specials include: Cross-Stage Partial Connection (CSP), SPP-blocks and SAM-blocks. The authors applied CSP to overcome the problem of duplicate gradient in DenseNet. In DenseNet each layer receives information from all preceding layers, which results in different dense layers learning copied gradient information. CSP separates feature maps from the base layer into two parts. One part will move through the dense block and a transition layer and the other part is combined with the first part after it has moved through the transition layer. CSP prevents an abundance of copied gradient information, while still allowing the network to reuse features. Spatial Pyramid Pooling (SPP) is applied in the form of an SPP-block. In this block max pooling is used with different kernel sizes. The feature maps from these different pooling stages are concatenated, which increases the receptive field. The authors also apply a modified version of Spatial Attention Module (SAM), which was orignally introduced by Woo et al. (2018). This modified SAM-block applies convolution and a sigmoid function to determine point-wise attention. All these improvements ensure that YOLOV4 is twice as fast but with comparable performance.
Swin Transformer: The introduction of the Shifted Window (Swin) Transformer (Z. Liu et al., 2021) has resulted in a paradigm shift, by moving from CNNs to transformer networks in object detection.

The Swin transformer first splits up the input image in non-overlapping patches, and converts them into embeddings. At every successive Swin Transformer block the number of patches is decreased, so each patch contains more pixels (see Figure 2.10a). In Figure 2.10b, two successive Swin Transformer blocks are shown, composed of window based local multi-headed self-attention (MSA) modules. A key concept is the shift of the window partition between consecutive self-attention layers, which allows for cross window connections.

While we are in the early stages of adopting Transformer for object detection, the results are very promising. Swin transformer achieved state-of-art on various object detection and semantic segmentation benchmark datasets.

2.3 Image anonymization

Redaction methods are techniques that conceal sensitive regions in an image by modifying or removing these regions. However, concealing these regions may remove information that is needed to understand the image. Hudson and Smith (1996) described this phenomenon as the privacy intelligibility trade-off. We follow the taxonomy proposed by Padilla-López et al. (2015) where the authors propose five redaction methods: image filtering, encryption, face de-identification, object removal and visual abstraction.

Image filtering: Image filtering, as the name suggests, applies a filter to an RoI to modify that region and conceal the privacy sensitive information. The most common image anonymization method is blurring where a Gaussian function that modifies pixels based on their neighbors is applied to (regions of an) image. Some examples are presented in C. Zhang et al. (2006), Frome et al. (2009) and Devaux et al. (2009). Pixelation is a method that divides an image into a grid, where each block takes the average value of the pixels inside that block. This method is commonly used to preserve the privacy of suspects on television, and also in the works of Boyle et al. (2000) and Kitahara et al. (2004).

Encryption: Another redaction method is to encode imagery data such that the original data becomes unintelligible, called encryption. This can be done by encrypting (parts) of the video bitstream (M. Yang et al., 2004), or by scrambling which can be applied to permute the spatial, frequency, or code-stream domain (Tang, 1997). The encrypted information can be retrieved by a key.

Face de-identification: This is a method that changes faces in such a way that human individuals nor face recognition software are able to recognise the face. These alterations can be done by applying aforementioned methods on parts of the face, such that the balance between intelligibility and privacy is kept. A common method is the K-same algorithm introduced by Newton et al. (2005). The K-same algorithm uses a distance metric to cluster similar faces, it then averages face components and applies this to all faces in the cluster. This algorithm was later extended by Gross et al. (2006) and Gross et al. (2009). A different approach is presented by Bitouk et al. (2008) where the authors propose a face image library, where the target face is swapped with a similar face from the library. More recent research involves the use of GANs for face de-identification (Kim and J. Yang, 2019; T. Li and L. Lin, 2019; Z. Ren et al., 2018).

Object removal: The process that is concerned with the removal of people and objects from an image is called object removal. This removal creates a gap in an image, which is filled using inpainting techniques. Inpainting consists in reconstructing missing parts in an image through the information of surrounding areas. Early methods applied texture synthesis (Efros and Leung, 1999) by using the texture information from one region to fill in the missing region. Bertalmío et al. (2000) proposed an automatic inpainting algorithm that propagates information from surrounding areas in the isophotes direction. Most of the early inpainting algorithms only use color information, L. He et al. (2011) proposed a method that takes into account both color and depth information. As in other research fields in AI, the introduction of Deep Learning dramatically increased the inpainting performance. Deep Learning-based methods such as GANs (Pathak et al., 2016; Iizuka et al., 2017; Demir and Unal, 2018) and Diffusion models (Saharia et al., 2022) have replaced traditional inpainting methods.

Visual abstraction: This method substitutes objects in images by their abstracted versions. Common abstractions could be a silhouette (Tansuriyavong and Hanaki, 2001), only maintaining shape but removing texture. A pseudogeometric model (D. Chen et al., 2007), able to preserve rich structure and motion information in videos. And the use of 2D models as mentioned in (Sadimon et al., 2010).

2.4 Generative models

A generative model is able to generate data following a data distribution, learned through training. Examples of generative models are Variational Autoencoders (VAEs) (Kingma and Welling, 2013) that learn to approximate the data distribution explicitly through maximum likelihood estimation, and Generative Adversarial Network (GANs) (Goodfellow et al., 2014) that are trained implicitly. Optimization of GANs requires finding a Nash Equilibrium which is more difficult than optimizing an objective function, used in VAEs (Goodfellow, 2017). Despite this difficulty, we focus on GANs, as their ability to generate realistic

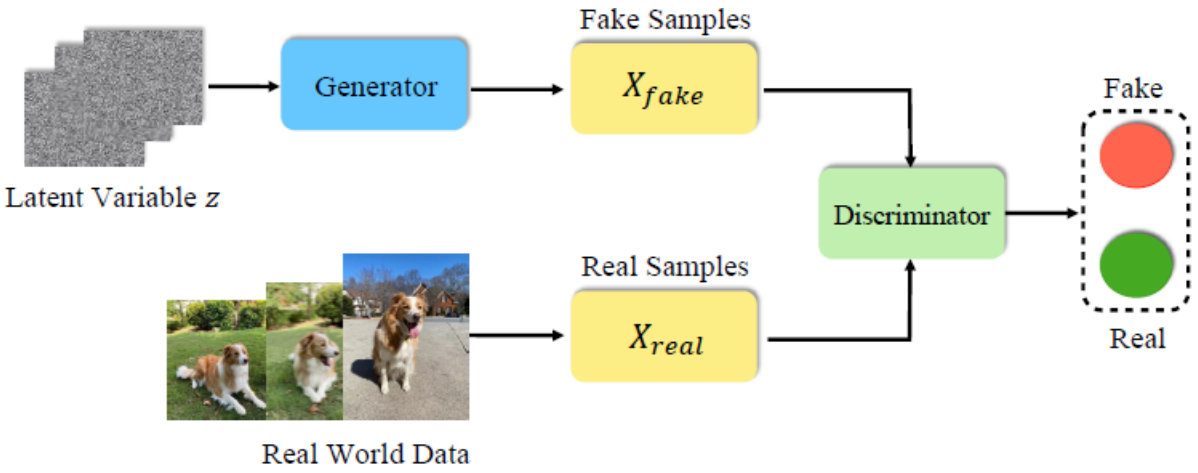


Figure 2.11: Basic architecture of GAN. Image from (Cai et al., 2021).

samples is better.

2.4.1 Generative Adversarial Networks

A GAN consists of two deep networks: a discriminator, with the goal to distinguish generated (fake) images from real images; and a generator, which aims to create images such that the discriminator classifies these generated images as real (see Figure 2.11). The two networks play a zero-sum game, also called minimax game. Here the discriminator learns by maximizing its correct predictions, while the generator learns by minimizing the discriminator's correct predictions. During training, the real images are sampled from the training set, and the generator creates images given a random fixed-size noise vector, drawn from a Gaussian distribution. Training is done in an alternating fashion, where one network is fixed and the other network is updated through backpropagation on its corresponding loss. Through this process, the generator learns to generate images that closely resemble the images from the training set.

In order to generate higher quality images, Radford et al. (2015) proposed Deep Convolutional GAN (DCGAN). Here the authors introduce the use of CNNs in both the generator and the discriminator, where the generator makes use of fractionally-strided convolutions, or transposed convolutions. Another improvement is the use of batch normalization in all but the first and last layers, and the elimination of fully connected layers.

A common problem in GANs is mode collapse. This is the problem that the generator only generates a small subset of possible outcomes. A possible solution to this problem is the introduction of Wasserstein loss proposed by Arjovsky et al. (2017). Wasserstein GAN (WGAN) is trained to minimize the Wasserstein distance between the real- and generated data distribution. This however can still lead to undesirable results due to weight clipping. Therefore, Gulrajani et al. (2017) proposed gradient penalty instead of weight clipping to improve performance.

2.4.2 Conditional Generative Adversarial Networks

Unconditional GANs generate data from the domain based on a random noise vector, this make the generation process inherently random, and inhibits control over the output of the GAN. Therefore, Mirza and Osindero (2014) proposed conditional GANs (cGANs) that extends the work by Goodfellow et al. (2014) to allow conditioning on the model. Such conditioning is done by adding auxiliary information, such as class labels. The addition of auxiliary information guides the model in terms of image generation.

The work by Isola et al. (2017) proposes conditional GAN (cGAN), which extends the work of Mirza and Osindero (2014) by allowing to condition upon some input image, for the task of image-to-image translation. Figure 2.12 shows the basic architecture of cGAN, where the conditional input is an image.

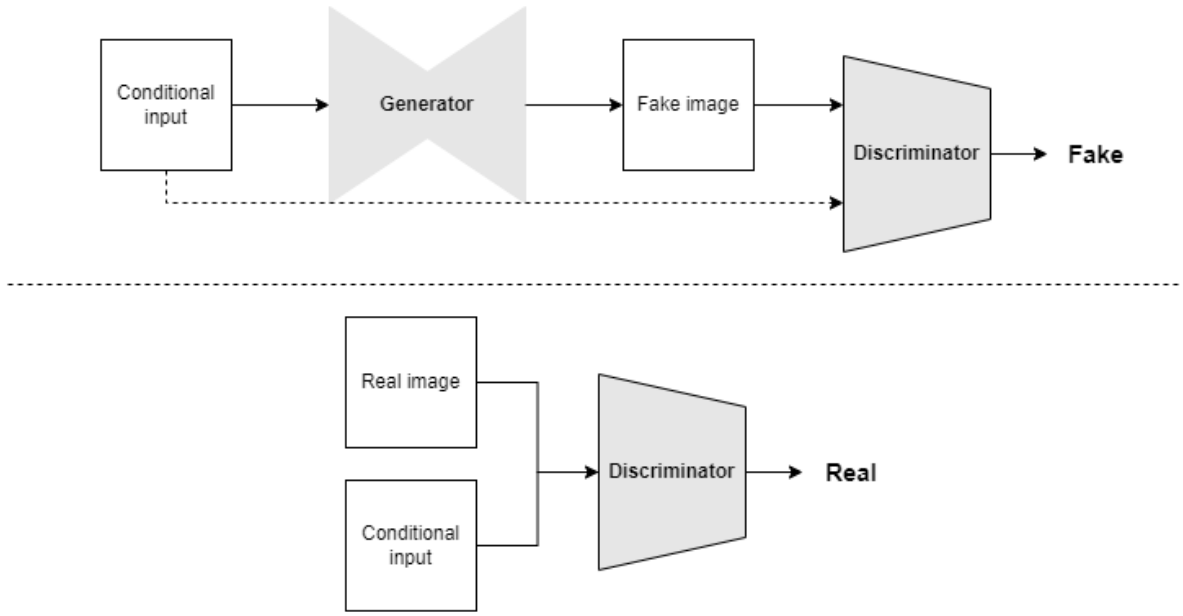


Figure 2.12: Basic architecture of a cGAN. The discriminator learns to distinguish based on a real/generated image and a conditional input. Both the generator and the discriminator observe the conditional input.

The discriminator is provided with both the conditional input and the generated image and the goal is to determine whether the generated image is a plausible transformation of the conditional input image. The generator is comprised of two parts, an encoder part and a decoder part. The encoder part receives the conditional image and compresses the image into a smaller representation, in order to abstract higher representations of the input image. The decoder takes the embedded input image, which is upsampled using transposed convolutions. The encoder and decoder are connected with skip connections in a "U-net" fashion. The skip connections offer the network the option to bypass parts of the encoder or decoder, this allows sharing of low-level information between input and output.

2.4.3 GAN for inpainting

Pathak et al. (2016) presented the first GAN-based image inpainting method. In this work the authors propose a context encoder, which is a CNN trained to apply inpainting to a region based on its surroundings. The network consists of a encoder and a decoder part that work the same way as in cGAN. A joint loss is used comprised of reconstruction and adversarial loss. For the reconstruction loss, L2 loss is used. The adversarial loss is based on GAN, where the context encoder replaces the generator, and competes with the discriminator to produce realistic output. The combination of L2 and adversarial loss is used because the former captures the overall structure of the missing region, whereas the latter tries to make the output appear more realistic. Although the network performs well when the missing region is square and is situated in the center of the image, performance deteriorates for random shaped missing regions or when the missing region is not in the center.

A major contribution to image inpainting by Iizuka et al. (2017) applies both a local and a global discriminator to capture more information from an image. The whole generated image is processed by the global discriminator, whereas the local discriminator only takes in the inpainted region. The combination of the two discriminators results in images that are both globally and locally consistent. The authors also propose a fully convolutional neural network with dilated convolutions. These dilated convolutions, proposed by F. Yu and Koltun (2015), obviate the use of fully connected layers, and therefor allowing input images of varying size. But more importantly, dilated convolutions use kernels that are spread out. This allows the network to compute each output pixel with a larger input area, while parameters and computation time remain the same. An overview of the architecture is shown in figure 2.13. Despite the fact that the network performs well, even with random shaped missing areas, the network struggles with

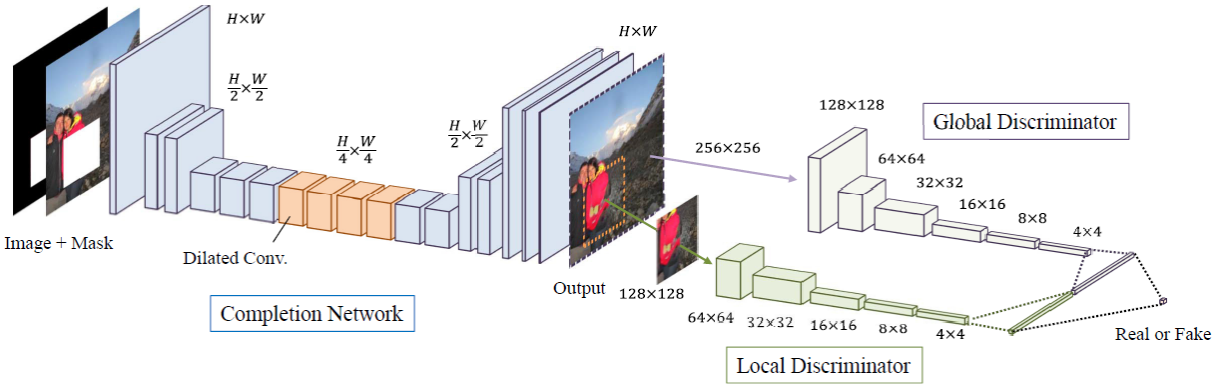


Figure 2.13: Overview of the architecture used by Iizuka et al. (2017). The network takes the image and the corresponding binary mask as inputs. Regular and dilated convolutions are applied. The image is the upsampled using transposed convolutions. The inpainted region is processed by the local and the whole image by the global discriminator. Combination of outputs is used to determine whether image is real or fake. Image from Iizuka et al. (2017).

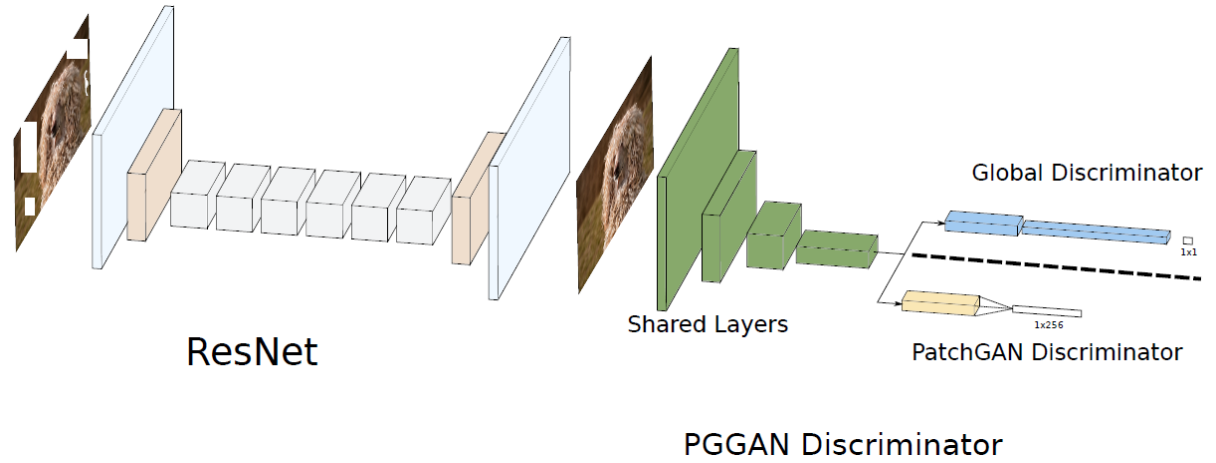


Figure 2.14: Overview of PGGAN. The image is compressed using dilated residual blocks and upsampled using interpolated convolution. The two discriminators share some convolutional layers and are the separated. The global discriminator outputs a single value. PatchGAN outputs a 1×256 dimensional vector which can be reshaped to a 16×16 matrix. Image from Demir and Unal (2018).

complex structures in an image.

Demir and Unal (2018) propose PGGAN, a novel inpainting network which combines residual learning (K. He et al., 2015) and PatchGAN (Isola et al., 2017). This network also utilizes two discriminators, a global discriminator and a PatchGAN discriminator (see figure 2.14). The two discriminators have a few shared convolutional layers and are then separated. The global discriminator looks at the whole image and outputs a binary value that decides whether the image is real or fake. The PatchGAN discriminator processes the image in a sliding window-fashion. The output is a matrix, where each value corresponds to the realness of a patch. The loss of both discriminators is combined with L1 loss, which ensures better pixel-wise reconstruction accuracy. The authors also combined dilated convolutions with residual learning to create dilated residual blocks. The residual blocks improve the gradient flow and dilated convolutions increase the receptive field. To overcome the checkerboard effect in the inpainted regions, the authors apply interpolated convolutions instead of transposed convolutions. The model performs well on missing regions in the center, but poorly on images with different colors and textures.

EdgeConnect by Nazeri et al. (2019) separates the inpainting process in two steps, where the model first predicts the edges and uses this information to fill the missing regions. The authors also employ partial convolutions, a technique proposed by G. Liu et al. (2018). This technique applies convolutions using a binary mask, such that missing pixels only get updated with information from valid pixels. In

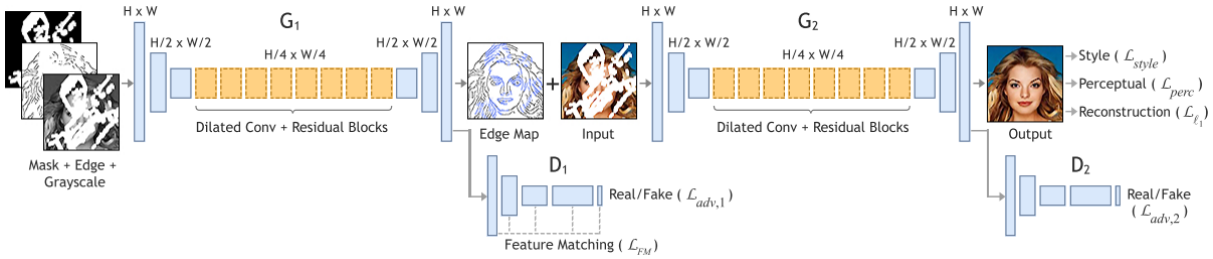


Figure 2.15: High-level overview of EdgeConnect. Image from Nazeri et al. (2019).

figure 2.15 the architecture of EdgeConnect is shown. The binary mask, edge map and grayscale image serve as inputs to the edge generator (G_1), which in turn outputs the completed edge map. G_1 is trained using standard adversarial loss and feature matching loss. The latter is acquired using the feature maps of the first discriminator (D_1). The L1 distance is computed between the feature maps of a ground truth and a generated image. The image generator (G_2) takes in the completed edge map and the masked RGB image and returns the inpainted image. G_2 is trained using 4 different loss functions. Standard adversarial loss, the loss of misclassifying an image. Perceptual loss, similar to feature matching loss but using feature maps of VGG. Style loss, computed as L1 distance between the gram matrices of the feature maps of VGG. And reconstruction loss, pixel-wise L1 distance between ground truth and generated images. EdgeConnect performs very well even on large random shaped missing areas.

The authors of DeepFill V2 (J. Yu et al., 2019) combine multiple techniques that we described previously with their Contextual Attention (CA) (J. Yu et al., 2018). The CA layer determines attention scores for each pixel. These scores describe the similarity between a pixel in the missing region and pixels that are known. The CA layer guides the inpainting process by determining what region to focus on for each pixel. The authors propose a gated convolution mechanism to improve the partial convolution operation. The idea here is that gated convolutions are able to learn a binary mask whereas partial convolution apply a rule-based binary mask. The gated convolution is achieved by adding a standard convolution layer with sigmoid activation before another standard convolution layer. Instead of a local or global discriminator, the authors combine PatchGAN with Spectral Normalization (SN) (Miyato et al., 2018), which they name SN-PatchGAN. Here SN is applied to all layers of the discriminators, the authors claim that SN makes the training more stable. SN-PatchGAN makes the use of perceptual loss redundant as patch information is already encoded in SN-PatchGAN. Based on qualitative and quantitative results DeepFill V2 is state-of-the-art GAN-based image inpainting method.

Most of the methods discussed above guide the inpainting process through the use of some prior, besides the input image. This has proven to be very effective and yields better results than methods that only use images to be inpainted as input. Furthermore, most methods also adopt multiple discriminators and generators and apply multiple loss functions. These techniques allow for more control over the model, and can be used to guide the model to a specific output. Finally, dilated convolution, gated convolution and contextual attention help the model to achieve better results by focusing on important parts of the image.

2.5 Metrics

A well known metric for GAN evaluation is the Fréchet Inception Distance (FID) (Heusel et al., 2017). This method uses the inception V3 (Szegedy et al., 2015). More specifically, the activations in the second last layer (before the output layer) are summarized as a multivariate Gaussian. This is done for both the real and generated set of images, between which the Fréchet distance is calculated. Another method is peak signal-to-noise ratio (PSNR) to determine the quality of the inpainted image. Although this is a good measure to evaluate the quality of a generated image or inpainted, it is not a measure for the level of anonymization.

To determine the performance of our anonymization algorithm, an appropriate metric should be selected. Although an abundance of studies on the anonymization of images exists, little research has

been conducted to assess the performance of these anonymization methods. A frequently used metrics in prior research is the use of human inspection. Here humans visually inspect a subset of the anonymized images (Uittenbogaard et al., 2019; Frome et al., 2009). Although this is an accurate metric, as humans are well able to determine whether the image is anonymized, it is a time-consuming task as well.

Another popular metric is the use of pre-trained object detection models, to determine whether a face or human body is present in the image. The authors of DeepPrivacy (Hukkelås et al., 2019) propose a face de-identification algorithm, whose performance is evaluated using face detection. For the evaluation the authors apply the Dual Shot Face Detector (DSFD) (J. Li et al., 2018). In their approach, DSFD is applied to examine the detectability of the generated faces.

Z. Ren et al. (2018) have a more extensive approach for the evaluation of their algorithm. The authors employ a combination of SSH (Najibi et al., 2017) and MTCNN (K. Zhang et al., 2016) for the detection of faces. In addition, the authors implement SphereFace-20 (W. Liu et al., 2017) for face recognition. This approach allows for the evaluation of an anonymized image, by determining both the presence and the recognizability of faces in the image. Although this is a more extensive metric, as we have established earlier, the anonymization of only the face is not sufficient to protect the privacy of human individuals in Street View data.

A clever, but rarely used method is person re-identification (Re-ID). The task of person Re-ID is to retrieve a person of interest across a set of images from non-overlapping cameras (Ye et al., 2020), or from the same camera in different occasions. The Re-ID can be performed based on soft-, non- or regular bio metrics, or a combination between those. Maximov et al. (2020) apply a person Re-ID algorithm, as a metric to evaluate their face de-identification cGAN. If the person Re-ID algorithm is unable to re-identify a generated image, given the corresponding real image, the anonymization algorithm is successful. Therefore, person Re-ID can be regarded as a suitable metric to determine the level of anonymization in an image.

Chapter 3

Methodology

In this chapter we propose methods for the generation of realistic anonymized images, and the evaluation techniques to determine the realism of the images generated by our model. In Section 3.1 we discuss our anonymization method and in Section 3.2 we describe our evaluation methodology.

3.1 Image anonymization

Previously we have mentioned the privacy intelligibility trade-off where concealing or removing regions from images can protect the privacy of persons in these images, but reduces the usefulness of the data. Removing or filtering a bounding box that encloses a person protects the privacy of said individual, however this also leads to a loss of information as parts of the background are also removed. Our method consists of two phases, the person removal phase and the image inpainting phase, visualized in Figure 3.1.

3.1.1 Method overview

Person removal

Considering that our research focuses on the anonymization of street view imagery, we use the Cityscapes dataset (Cordts et al., 2016) which contains images of people walking through various cities. In addition, this dataset provides high quality pixel-level annotations for various classes. These are described in more detail in Section 3.3. Our method requires the availability of this semantic information for both the person removal phase, as well as the inpainting phase. In this first phase we replace all person annotated pixels with a constant value. This replacement results in an incomplete image, where all person annotated pixels are converted to black. We will refer to these black regions as masks. This method effectively

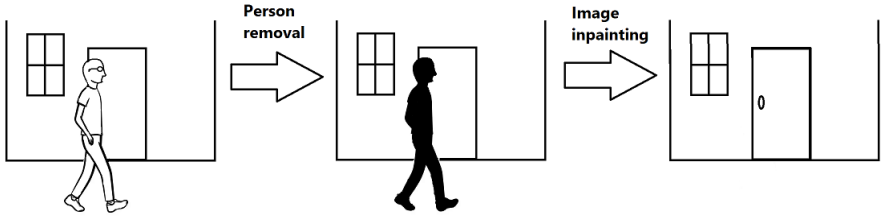


Figure 3.1: Our proposed method for image anonymization, with two stages: 1) Remove all persons from the image. 2) Generate new pixels in the missing regions left by the person removal stage.

removes persons while it leaves their immediate surroundings intact. Therefore, person segmentation is preferred over bounding boxes, as bounding boxes remove personal identifiers equally well, but remove more of the background information, making it more difficult to restore the image.

Image inpainting

The person removal phase delivers an incomplete image with masks that follow the silhouette of the removed persons. In the second phase we apply a GAN model to restore these masked regions using the information still present in the image. We implement two models based on Isola et al. (2017)'s pix2pix model. The first model is a single-stage model that only uses the incomplete RGB image for the inpainting process. The second is our SemGAN, which is a two-stage model that combines the incomplete RGB image and the semantic information to complete the image. This is novel as we introduce semantic information as a prior. We combine the incomplete RGB image with semantic information, where the latter is meant as guidance for the inpaiting process. This is especially helpful in street view data, as most images contain multiple distinct objects or regions with different semantic labels.

The semantic information can constrain the generator to only generate regions that are consistent with the semantic labels belonging to these regions.

The training process of SemGAN consists of two phases, the training phase and the fine-tuning phase. In the training phase, the GAN models learn to generate and complete missing regions based on the pattern of the masks in the training images. During the training phase, the model learns how to fill in rectangular patches that are missing, while during the fine-tuning phase, the model's ability to inpaint is extended to more complex shapes, such as those resembling human figures. We made this distinction because the rectangular inpainting task provides a more consistent way to evaluate the model. Each image contains an equal number of masks that are roughly the same size, so the varying scores are not due to differences in mask sizes but to other factors, such as the complexity of textures or the number of semantic classes.

The fine-tuning phase is required because of the difference in shape and complexity between rectangular masked images and human silhouette masked images. If the masks are rectangular, the models will be biased towards generating and completing missing regions in that shape. Inpainting input images with different shaped masks pose a challenge for these models, as these input images have a different underlying distribution than the images on which the models are trained on. As a result, the models may not be able to effectively learn the appropriate features to inpaint the person silhouette masks, which leads to noisy and artifact-filled results. To address this issue and allow the GANs to better learn to inpaint human silhouettes, it is necessary to fine-tune the models on masks in the shape of human silhouettes. This will help the models to adapt to the specific shapes and features of human shaped silhouettes and result in improved inpainting results when presented with similar masks.

Important to note is that the training process requires ground truth images, showing what the scene actually looks like. Therefore we adopt two masking algorithms. During the training phase we apply a masking algorithm that creates four rectangles at random locations. The height of these rectangles is in the range of 59 and 79 and the width ranges from 32 to 42, both height and width are uniformly sampled from their respective interval. We have chosen to do this as it mimics the pattern of individuals being near and far from the camera. In the fine-tuning phase we apply a masking algorithm that creates masks that follow the shape of human silhouettes. How this algorithm works is explained in Section 3.3.

The models are trained on the rectangle mask training dataset and are fine-tuned on the person mask training dataset. During the training procedure we employ a learning rate $\alpha$of 2×10^{-4}, as suggested in Isola et al. (2017), gives us the best results. All three models are trained for 10 epochs. For the fine-tuning procedure, α was decreased to 2×10^{-5}, which is 10 times smaller than the original value. The number of epochs was reduced to 5 for the pix2pix model and 7 for SemGAN and SemGAN-GT. Fine-tuning our models for less than 7 epochs led to a large performance gap between our models and pix2pix. This was reflected in more noisy inpainting results as well as worse scores across all metrics.

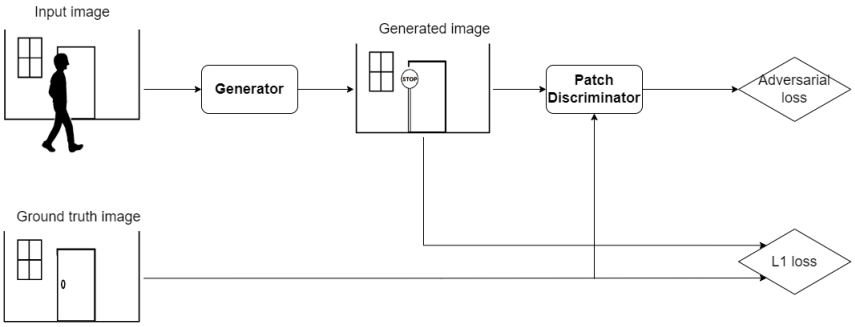


Figure 3.2: Isola et al., 2017's pix2pix model illustrated for our image anonymization procedure. This base model employs a generator and a PatchGAN discriminator, and uses the adversarial and L1 loss to train the model.

3.1.2 Base model

Our base model is adapted from Isola et al. (2017), introduced in Section 2.4.2. This architecture is visualized in Figure 3.2. The authors present their model as an Image-to-Image translation model, capable of converting an image into another. Image inpainting can be regarded as such a process, where we convert an incomplete image into a complete one by utilizing the information from the remaining pixels. This is an iterative learning process between the generator and the discriminator. The generator learns to fool the discriminator by generating fake images that look realistic, while the discriminator learns to classify fake and real images correctly.

Conditional GAN Objective: The objective function of the cGAN is displayed in equation 1. This function, also referred to as the adversarial loss, is a combined function of the discriminator D and the generator G, where x is the conditional input (incomplete image), y is the ground truth image and z is the random noise (through dropout in the generator during training and inference). The left term in this equation represents the performance of the discriminator in classifying the real and fake images correctly, whereas the right term describes the generator's ability to generate realistic images.

$$\mathcal{L}_{cGAN}(G, D) = \mathbb{E}_{x,y}[logD(x, y)] + \mathbb{E}_{x,z}[log(1 - D(x, G(x, z))] \tag{1}$$

L1 loss: The authors also implement L1 loss and combine this with the adversarial loss. The function, shown in equation 2, calculates the pixel-wise distance between two images, the ground truth and the generated image. On its own, minimizing the L1 loss would result in blurry images as the function takes the average of a group of pixels when uncertainty exist about the location of edges. Therefore the L1 loss is combined with the adversarial loss. This forces the generator to generate images that look both realistic and also to be near the ground truth. This is particularly important for inpainting as we want to minimize the loss of information. Although the L2 loss follows the same principle and therefore could be substituted for L1, the latter is preferred as it encourages less blurring (Isola et al., 2017).

$$\mathcal{L}_{L1}(G) = \mathbb{E}_{x,y,z}[\|y - G(x, z)\|_1] \tag{2}$$

Discriminator: The conventional technique to classify an image as real or fake is a global discriminator. The discriminator is a fully convolutional network, capable of learning and distinguishing between features of real and fake images. The output is a single value, i.e. the probability that the entire input image is real.

In pix2pix, the authors adopt a variation of this discriminator, the PatchGAN classifier. Instead of "looking" at the entire image, it divides an image into $N \times N$ patches and assigns a probability to each patch, representing the expectation that a patch comes from a real (close to 1) or a fake image (close to 0). This is especially useful for inpainting as this allows the PatchGAN to assess the transition between the generated parts and the original image, as an unrealistic boundary can be classified as fake.

In PatchGAN, each patch is independent from the other patches. This implies the independence

between pixels that are separated by more than a patch diameter. Therefore, the authors argue that PatchGAN can be understood as a form of texture loss, forcing the generator to create coherent texture.

Generated and ground truth images serve as an input to the discriminator. We condition both the generated and the ground truth image on the input image, i.e. the incomplete image. This is achieved by the channel-wise concatenation of the incomplete and corresponding complete image. By conditioning on the input image, the PatchGAN discriminator is able to use the information in the image to make more informed decisions about whether the patches are real or fake.

Generator: The generator adopts an encoder-decoder architecture, where the encoder processes the input image, on which we condition, and extracts high-level features from it. The decoder network then uses these features to reconstruct the output image through the utilization of transposed convolutions. We define the final objective in equation 3 to achieve the above mentioned goals of fooling the discriminator and minimizing the L1 loss.

$$G^* = \arg\min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_{L1}(G) \tag{3}$$

An important technique used in the generator are skip connections, which allow information to flow between encoder and decoder, bypassing intermediate layers. During the process of inpainting, the incomplete input image and the generated output image often share a significant amount of information, particularly with regard to the background. Skip connections should allow this information to flow directly from encoder to decoder.

However, we have found better results when we use the background information from the original image directly. Therefore, the generated output is a combination of the image generated by the generator, $\hat{G}(x, z)$, and the original image y. Equation 4 defines the inpainted image as $G(x, z)$. Where M is the binary mask with ones at each location a pixel is dropped and zeros for the input pixels.

$$G(x, z) = M \odot \hat{G}(x, z) + (1 - M) \odot y \tag{4}$$

3.1.3 Segmentation model

In the work of Xiao et al. (2019), the authors reveal the human approach in picture restoration where a separation between content and style is made, dividing the process in less complex subproblems. Hence, incorporating semantic information into the image inpainting process is a logical step. A semantic segmentation map contains pixel-level semantic information, including the layout, category, location, and shape of objects in a scene, which can help to learn the texture variations across different semantic regions.

Our segmentation-based model SemGAN is an extension of the pix2pix model described in the section above, which only used an RGB image as the conditional input. The input to SemGAN is a combination between the incomplete RGB image and a completed semantic segmentation map, which are concatenated channel-wise. To acquire the complete segmentation map of the image, we adopt another pix2pix-based GAN that is specialized in the completion of semantic segmentation maps. This results in a pipeline with two stages, illustrated in Figure 3.3. The first stage involves the completion of the semantic segmentation map. This completed segmentation map is then used as a prior for the SemGAN model. The segmentation map is transformed to a one-hot encoding of the labeled data. Using one-hot encoding allows the GAN model to better distinguish between different classes in the segmentation map, as each class is represented by a unique binary vector. In the second stage we use this completed segmentation map to inpaint the RGB image. The GANs are trained separately and combined after training.

$$\mathcal{L}_{CE}(l, \hat{l}) = -\sum_{i=1}^{i=N} l_i \cdot log(\hat{l}_i) \tag{5}$$

The first GAN is trained using a combination of the adversarial loss (equation 1) and the cross entropy

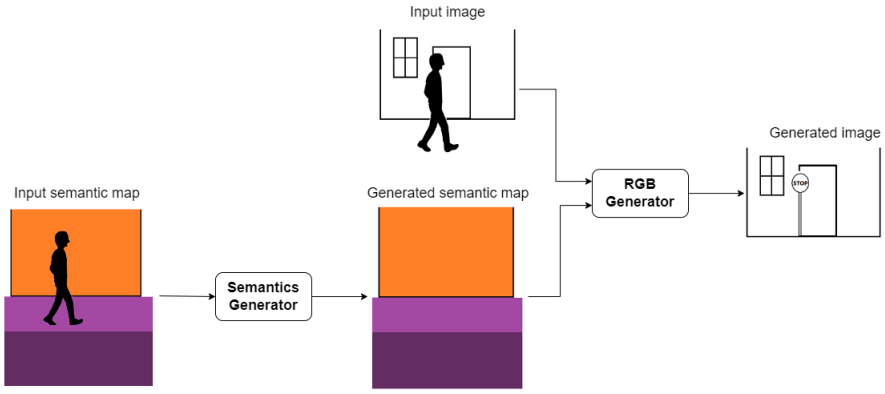


Figure 3.3: Our proposed pipeline to improve the inpainting process by including semantic information. The first step involves the completion of the semantic segmentation map. In the second step the generated semantic map is combined with the incomplete RGB image as a prior for the RGB generator.

loss as described in equation 5, where l is the truth label and $\hat{l}$ is the probability for the predicted label for class i. Equation 6 is the reformulation of the final objective for the semantics generator, where γ is a weighting parameter, similar to λ.

$$G^* = \arg\min_G \max_D \mathcal{L}_{cGAN}(G, D) + \gamma \mathcal{L}_{CE}(l, \hat{l}) \tag{6}$$

3.2 Metrics

In this Section, we examine the techniques employed to assess the level of realism of the anonymized images generated by our method. We apply three different metrics: L1 distance, peak signal-to-noise ratio (PSNR) and structural similarity index measure (SSIM). These methods are all good proxies for realism as they are able to measure reconstruction and visual quality. This is opposed to Fréchet Inception Distance (FID) (Heusel et al., 2017), another common metric in GAN evaluation, but not suitable for inpainting. This metric compares global statistics of the feature representations, but does not consider inpainting quality. In addition, FID is not sensitive to distortions, which are very common in inpainted regions (Uittenbogaard et al., 2019).

Although we use the same metrics for the rectangle and person inpainting tasks, the implementation is different.

3.2.1 L1 distance

The most straightforward method and often used in inpainting research is the L1 distance metric, which can be used as a measure for image (dis)similarity. This is a suitable metric for inpainting as we want to measure the reconstruction quality, rather than its ability to produce diverse content through inpainting.

The L1 distance is defined as the sum of the absolute differences between corresponding elements in two images, its mathematical representation is shown in equation 7, where $I1$ and $I2$ are two different images. The closer the value of L1 gets to zero, the more alike two images are, with larger values indicating more dissimilarity.

As discussed in the Section above, L2 could serve as an alternative to L1. However, we prefer L1 as this is less sensitive to outliers, which are still very common in the generated images. Although L1 lacks the ability to measure texture, which is an important aspect when it comes to realism, it is still a good proxy for realism.

$$L1 = \sum_{x,y} |I1_{x,y} - I2_{x,y}| \tag{7}$$

3.2.2 Peak signal-to-noise ratio

Another common metric for image inpainting is the peak signal-to-noise ratio (PSNR), a full reference metric that compares two images. In this context, the original data can be thought of as the "signal", and the error introduced by the inpainting process is considered the "noise", the ratio is calculated in decibel form. The PSNR is typically computed on a decibel scale, which is a logarithmic scale, because the signals being compared can have a very large range of values.

PSNR is calculated using the maximum possible pixel value of the image, represented as MAX^2 and the mean squared error (MSE), the formula is shown in equation 8. The higher the PSNR value, the less distortion there is between the signal and the noise, indicating that the inpainting process has generated a high-quality result.

PSNR also has its limitations as shown by Z. Wang et al., 2004. The authors alter an image using different types of distortions, where the images clearly have different visual quality, but observe identical PSNR values for the distorted images. It may not be the best metric for realism as it only measures pixel differences and does not consider factors such as the quality of the texture or visual artifacts. However, since PSNR is, to some extent, able to measure the quality of an image, simple to calculate, has clear physical meanings and is used as a benchmark in many inpainting studies, we chose to employ PSNR as a metric.

$$PSNR = 10 \log_{10} \left(\frac{MAX^2}{MSE} \right) \tag{8}$$

3.2.3 Structural similarity index measure

Structural Similarity Index (Z. Wang et al., 2004) (SSIM) is a good metric for inpainting because it measures the structural similarity and visual quality between the original image and the inpainted image, resulting in a value between +1 and -1. A value close to +1 means a high similarity between images and a value close to -1 indicates a large difference between images. This metric takes into account the way the human visual system perceives images and considers factors such as luminance, contrast and structure of the image.

SSIM is an effective metric for evaluating the realism of inpainted images because it considers both the pixel values and the structure of the image. It evaluates the similarity between the ground truth and inpainted image by comparing patterns of pixels in local regions. This method is grounded in the idea that the human eye is more sensitive to variations in the structure of an image rather than the variations in the pixel values. In addition, SSIM is relatively robust to image compression and noise, which makes it more suitable for image inpainting applications.

3.2.4 Difference in metrics

We adopt two different methods for the calculation of PSNR and SSIM of an image, patch-wise and image-wise comparison. For the task of rectangle inpainting we adopt the former, while we use the latter to analyze the person inpainting results. In the patch-wise evaluation method, the two metrics are calculated between the inpainted patches and their corresponding ground truth patches, and then averaged across all patches of the image. The image-wise method, as the name suggests, consists of calculating the metrics between the entire image with inpainted patches and its ground truth counterpart. The main reason for this differentiation is that the locations of the masks are known for the task of rectangular mask inpainting, whereas these are unknown for the person masks.

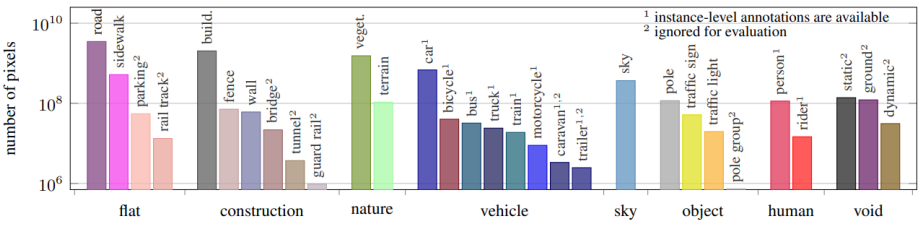


Figure 3.4: Number of finely annotated pixels per class (y-axis) with corresponding categories(x-axis). Figure from Cordts et al. (2016).

Since the methods have a different approach, the patch-wise scores and image-wise scores vary for the same image. The reason for the difference in PSNR and SSIM scores for the two methods is that they are evaluating different aspects of the inpainting performance. The image-wise method compares the entire inpainted image with the ground truth image, while the patch-wise method only evaluates the inpainted patches and their corresponding ground truth patches. Therefore, this method is more sensitive to errors in the inpainted patches, as it focuses solely on these regions, while the image-wise method takes into account the entire image, including the surrounding areas that are similar between inpainted and ground truth image. As a result, this method can mask the errors in the inpainted patches, leading to a higher PSNR and SSIM score compared to the patch-wise method. SSIM is especially sensitive to this issue, as calculating the structural similarity between the entire ground truth and inpainted image result in only very small differences in SSIM values (all values close to 1). Therefore, the image-wise comparison is not suitable to determine the quality of the inpainted patch.

3.3 Dataset

We train, test and evaluate our model on the Cityscapes dataset (Cordts et al., 2016). It contains street view images from 50 different cities, with high quality pixel-level annotations of 5,000 frames in addition to a larger set of 20,000 weakly annotated frames. However, we will only use the high quality annotations and their corresponding images. Cityscapes consists of 30 classes (see Figure 3.4), and contains a lot more annotated pixels than other similar datasets such as the KITTI Vision Benchmark Suite(Geiger et al., 2013).

We focus on the anonymization of images, therefore we are highly attentive about the number of persons that occur in the dataset. In roughly 1% of the images persons appear, with a total of 24400 persons throughout the dataset. Hence, some images do not contain persons and others contain multiple.

The pre-processing procedure consists of the following steps: resizing, masking and normalizing. The images are resized using a nearest neighbor approach that reduces the images to 256×512, speeding up both the training and inference time. As we described, we employ two masking algorithms. The first algorithm, randomly places a number of black rectangles in the image and the second creates black random silhouettes at random locations in the image. We normalize the RGB values between -1 and +1 to stabilize and speed-up the training process.

For the task of person inpainting, we have modified the training and validation datasets to exclusively include images without persons. The images in these datasets are masked with silhouettes of persons, acquired from the Cityscapes images that do contain persons. In order to obtain masks that are large enough, we have only selected those that contained more than 3500 "person" labeled pixels. The construction of the fine-tuning training dataset is essential for obtaining ground truth images that are required for the training procedure. Likewise, the creation of the fine-tuning validation dataset is crucial for assessing the performance of the inpainting model using these ground truth images. The fine-tuning training dataset contains 649 images, while the validation dataset consists of 98 images.

Chapter 4

Results & Discussion

In this chapter, we present the results of our research on using GANs for the task of inpainting people in images. We trained three GAN models using Cityscapes, a streetview imagery dataset containing images of people, as well as semantic segmentation maps for the corresponding images. The dataset consists of a training (2975 images), validation (500 images) and test split (1525 images). However, we will only evaluate our models on the validation split as the semantic information for the test subset is not public.

The three models compared are the pix2pix model, our two stage SemGAN model using the predicted semantic map, and SemGAN-GT that uses the ground truth semantic map. The results in Table 4.1 report the average score and standard deviation over a single independent training run.

Model	L1 ↓	SSIM ↑	PSNR ↑
pix2pix	4204 ± 1730	0.4093 ± 0.1355	21.23 ± 3.54
SemGAN*	4012 ± 1776	0.4549 ± 0.1480	21.89 ± 3.78
SemGAN-GT	**3820** $\pm$ 1661	**0.4630** $\pm$ 0.1443	**22.21** $\pm$ 3.66

Table 4.1: L1, Structural Similarity Index Measure (SSIM) and Peak Signal-to-Noise Ratio (PSNR) outcomes for the validation dataset across all models for the rectangular masked image inpainting task. ↓ lower is better. ↑ higher is better. *IoU of predicted semantics was 0.47.

4.1 Rectangular masks

In this section we evaluate and compare our models for the task of inpainting rectangular masked images. Rectangular mask inpainting is the benchmark task, as it is relatively straightforward and allows for a patch-wise quantitative evaluation of the models' performance. The patch-wise method allows us to more accurately evaluate the inpainting performance, as it only considers the inpainted patches instead of the whole image. Therefore, the score reflects the true inpainting performance of the different models. In addition, all images in the dataset contain the same number of masks of (roughly) the same size, making it easier to compare inpainting results across the dataset. The assignment of masks to random locations ensures that the models are challenged to inpaint a variety of objects and structures, allowing the models to generalize better.

The results from Table 4.1 provide an insight in the performance. As a reference and to better interpret the metric scores, we have included Figure 4.1. The left image depicts masked patches with random values, created by an untrained GAN with randomly initialized weights. The right image contains masked patches with a fixed value, resulting in black patches. Although the patches in both images appear unrealistic, they report significantly different scores. The inpainted result leads to better L1 and PSNR metrics, but results in a significantly worse SSIM score compared to the image on the right. These conflicting results indicate that relying on a single metric to evaluate the realism of an image containing patches may not be sufficient and may lead to inaccurate conclusions.

Figure 4.1: Images and their corresponding metric scores. The left image is generated by an untrained GAN model, whereas the right image is a standard input image to our GAN models.

4.1.1 Quantitative analysis

The scores in Table 4.1 indicate that our SemGAN-GT model has superior performance on all three metrics. The improved performance of both SemGAN and SemGAN-GT compared to other models suggests that incorporating semantic information as a prior could be a valuable approach in image inpainting. The results suggest that the inpainted patches generated by SemGAN and SemGAN-GT are more similar to the ground truth in terms of pixel intensity, as demonstrated by its higher L1 scores. The higher SSIM scores for our semantic aware models indicate that the inpainted patches produced have a higher level of structural similarity to the ground truth than pix2pix. An improved PSNR score for both of our models indicates that the patches inpainted by pix2pix have higher levels of distortions and noise. We have included the Intersection over Union (IoU) to indicate the overlap between the predicted and ground truth labels. For SemGAN the IoU is 0.47. This indicates that the size of the intersection, which represents the correctly predicted labels, is approximately half the size of the union, which represents the total occurrence of classes in both the prediction and ground truth.

Model	Difficulty	L1 ↓	SSIM ↑	PSNR ↑
pix2pix		731 ± 678	0.5694 ± 0.2433	24.89 ± 6.26
SemGAN	Easy	734 ± 695	**0.5855** ± 0.2500	**24.90** ± 6.32
SemGAN-GT		**725** ± 608	0.5815 ± 0.2494	24.86 ± 5.98
pix2pix		938 ± 800	0.4900 ± 0.2466	22.77 ± 6.27
SemGAN	Moderate	935 ± 803	0.5085 ± 0.2537	22.95 ± 6.35
SemGAN-GT		**905** ± 737	**0.5100** ± 0.2541	**23.23** ± 6.15
pix2pix		1459 ± 860	0.2995 ± 0.1969	18.16 ± 4.90
SemGAN	Hard	1375 ± 816	0.3236 ± 0.2050	18.64 ± 4.94
SemGAN-GT		**1232** ± 677	**0.3611** ± 0.1967	**19.53** ± 4.53

Table 4.2: Comparing model performance based on difficulty level of masked patch on the validation set. The dataset consisted of 736 easy, 332 moderate, and 932 hard patches.

Different textures in an image make the inpainting task increasingly difficult. Table 4.2 presents the results obtained for varying levels of difficulty, separating the masked patches into three categories: easy, moderate, and hard. The difficulty level of the masked patches in the dataset is defined based on the number of semantic labels they overlap. Easy masked patches overlap one or two semantic labels, moderate masked patches overlap three semantic labels, and hard masked patches overlap four or more semantic labels. The results follow the pattern from Table 4.1, as SemGAN-GT has the overall best performance. The categorization of the masked patches into different difficulty levels demonstrates the trend that an increase in the number of semantic labels, and thus in diverse textures in a patch, leads to a more challenging inpainting task. Additionally, we see the difference between the models using semantic information and pix2pix increase as the difficulty increases. This indicates that semantic information becomes more important as the number of object classes (and the the amount of different structures/textures) in the masked patches increases.

The complexity of the inpainting task is not solely determined by the number of semantic classes present in a masked patch, but also by the diversity of structures, texture and colors within the object classes. In particular, object classes like cars or buildings, which exhibit a wide range of colors and structures, present a greater challenge for inpainting as opposed to simpler object classes such as sidewalks

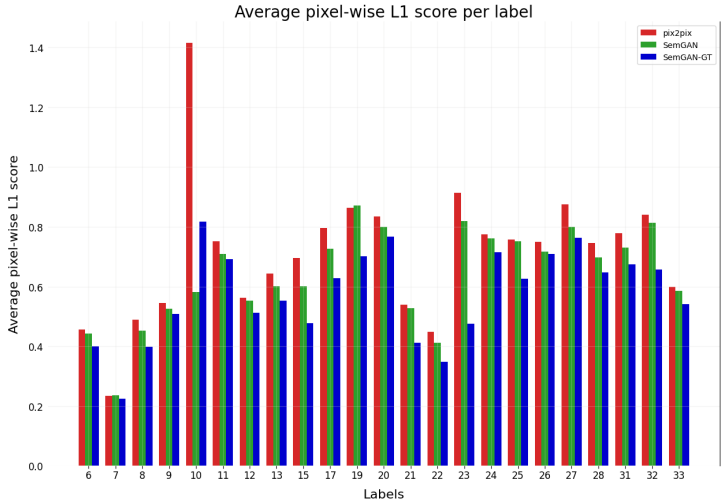


Figure 4.2: This figure shows the average L1 score for the inpainted patches, per label. The labels and their corresponding objects can be found in Table 4.3. A lower score indicates better performance.

or roads with fewer variations in color and structure. Figure 4.1 shows a bar plot of the average pixel-wise L1 score per label. For this graph we have only selected the classes that occur in the confusion matrix in Figure 4.7. See Table 4.3 for the labels and their corresponding objects.

The first thing than stands out in the graph is the poor performance of pix2pix in the inpainting of rail tracks (10) and sky (23), as demonstrated by its significantly higher L1 score. More specifically, when it comes to rail tracks, pix2pix shows an L1 score that is twice as high as that of SemGAN and SemGAN-GT, whereas its L1 score for sky is twice that of SemGAN-GT. The masking of these object classes in the training dataset is scarce. The dataset contains a limited amount of rail tracks and a low frequency of masked sky patches compared to other object classes. While the sky is not scarce in the dataset, masked sky is scarce due to the masking algorithm's constraints. This scarcity could make it difficult for the pix2pix model to inpaint these object classes. SemGAN and SemGAN-GT achieve improved performance on object classes through the use of semantic information, enabling effective inpainting of less prevalent object classes. This could be attributed to the models' ability to learn the relationship between labels and the visual attributes of object classes, leading to effective identification and reconstruction.

Another pattern we find in this graph is the superiority of the models that use semantic information over pix2pix, where SemGAN-GT is best among the two. This shows that semantic information not only improves inpainting performance in mixed scenes with multiple different object classes, but also in the inpainting of homogeneous scenes, where there is only one object class present in the patch.

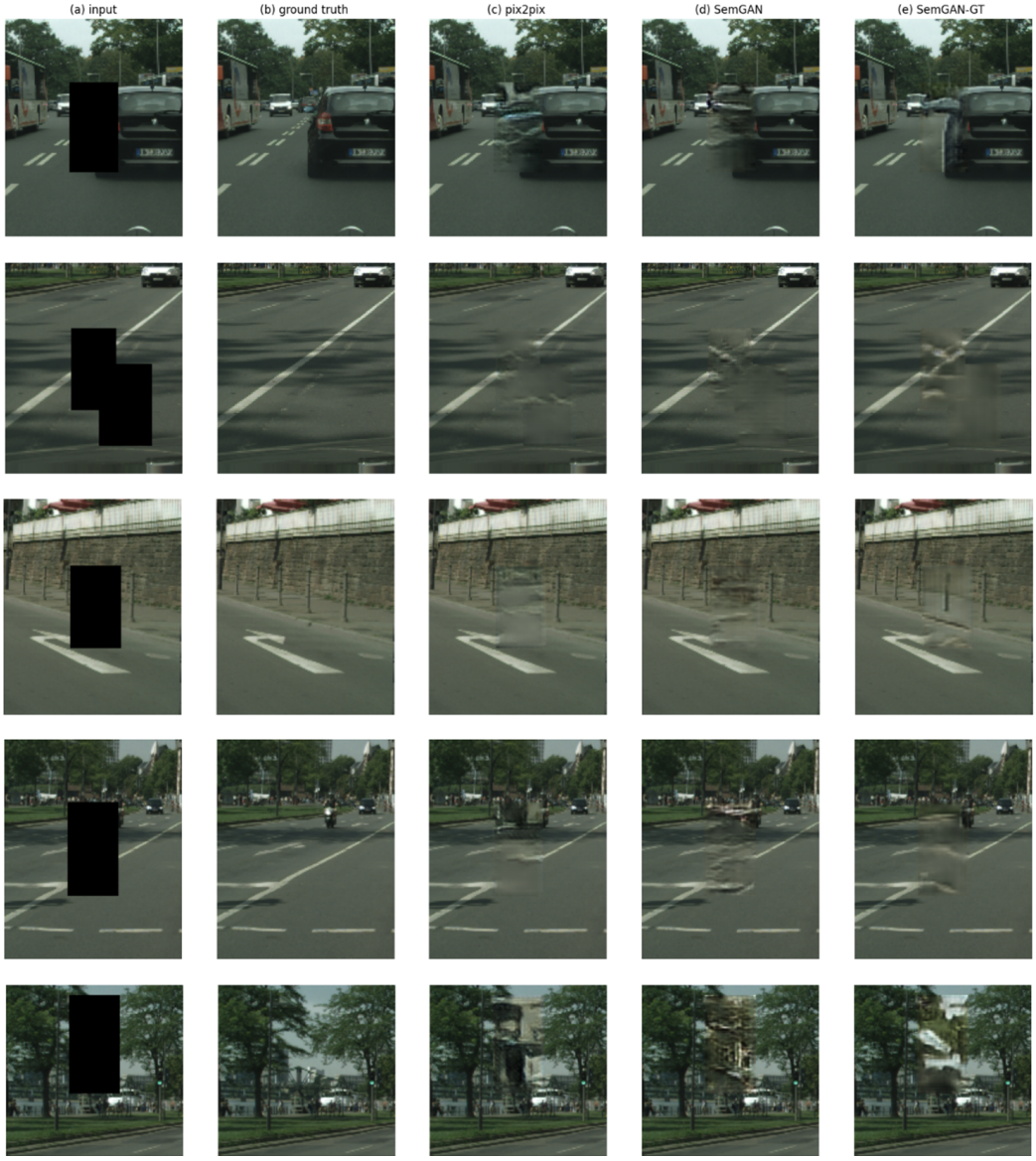


Figure 4.3: Ground truth images (b), compared with inpainted images that are generated by: pix2pix (b), Semgan (c), and SemGAN-GT (d) using the input images (a).

4.1.2 Qualitative analysis

The following section aims to visually compare and qualitatively evaluate the realism of the validation set images that are inpainted by our three models. Figure 4.3 displays the inpainting results, with the original images presented in the first column and the images inpainted by each of the models in the subsequent three columns. The input image in the top row contains a patch that partly masks the car. We see *c* and *d* struggle to effectively recreate the car. Pix2pix generates significant noise and visual artifacts, while SemGAN shows only a slight improvement, as the inpainted patch still contain noticeable noise. On the other hand, SemGAN-GT demonstrates its successful performance in reconstructing both the car and the road, along with inpainting the background traffic, as opposed to the other evaluated models. When inspecting the results in the second row we see that the models are able to inpaint the grey asphalt, but all struggle to complete the white line. Image *b* in the third row shows a row of poles, where one of the poles is masked by a patch. Pix2pix and SemGAN are both able to inpaint the road, sidewalk and part of the wall. This displays a common problem, where the GAN models are not able to recognize and reconstruct these patterns. SemGAN-GT is able to also reconstruct the pole, as this information is included in the ground truth semantic segmentation map. The fourth row again demonstrates the

superior performance of SemGAN-GT. Pix2pix and SemGAN are able to model the grey asphalt, but lack the ability to complete the white lines. Both models are unable to effectively inpaint the masked part of the scooter, as both reconstructions contain a lot of noise which makes it challenging to recognize the object. SemGAN-GT is able to clearly model the borders between the semantic classes, as grass, road and scooter are clearly separated. The same holds for the bottom row where SemGAN-GT is able to distinguish between sky and tree while the other models fail to correctly inpaint the patches.

There are several common challenges to image inpainting, as well as some challenges unique to each model. The most apparent problem, the impact of misclassifications in the semantic map on the inpainted image will be discussed in Section 4.2. One of the common difficulties we frequently observe is the edge problem, where the model is not able to complete the lines or edges masked by a patch, this is depicted in Figure 4.4. The top row shows the inability of all three models to complete the pavement markings, despite partial information about the markings still being present in the masked image. In the second row it seems that the models attempt to complete the center marking as the inpainted patches contain colors that closely resemble the colors of the center marking. However, it is more plausible that the models employ information from the sunny pavement to fill in the patch, as they conform to the distribution of both shaded and sunny regions. This suggests that the models utilize data from both areas to inpaint the patch, and again fail to complete the pavement markings. The bottom two rows depict the inpainted images that display edges for curbs and poles, which are present in the semantic information (as opposed to pavement markings). Both pix2pix and SemGAN fail to complete these structures, whereas SemGAN-GT demonstrates significantly better performance in completing the interrupted edges. These findings suggest that edge information present in semantic segmentation maps has a significant impact on image inpainting. The results also demonstrate that errors in the inpainted semantic segmentation map can propagate to the subsequent image inpainting task, leading to further errors in the final output. The edge problem is not limited to just these objects. It is also observed in trees, lampposts, and other elongated, thin objects.

Figure 4.4: Ground truth images (b), compared with inpainted images. This figure visualizes the edge problem. The upper two patches show incomplete pavement markings, while the bottom two patches depict the inability of the models to complete curbs or poles.

The second challenge we observe is that in some occasions the inpainted patches do not match the color distribution of the input image. This is a common problem called "color bleeding" (T. Wang et al., 2021). Two of these cases are depicted in Figure 4.5. In the top row pix2pix fails to match the color distribution of the image, where the inpainted patch contains much more blue than its surrounding area. The other models also fail to match the color distribution, as the overall patch contains a shade of brown. This same problem holds for the bottom row, where all patches contain similar mismatches in color distribution. Considering that all of the models used in the experiment encounter the same problem while working with the same set of images, it leads us to believe that the issue lies in the Cityscapes dataset. The dataset is composed of images captured throughout the year, which may result in variations in lighting conditions across the different dataset splits. Therefore, it can be suggested that these variations might be causing the problem in the models' performance.

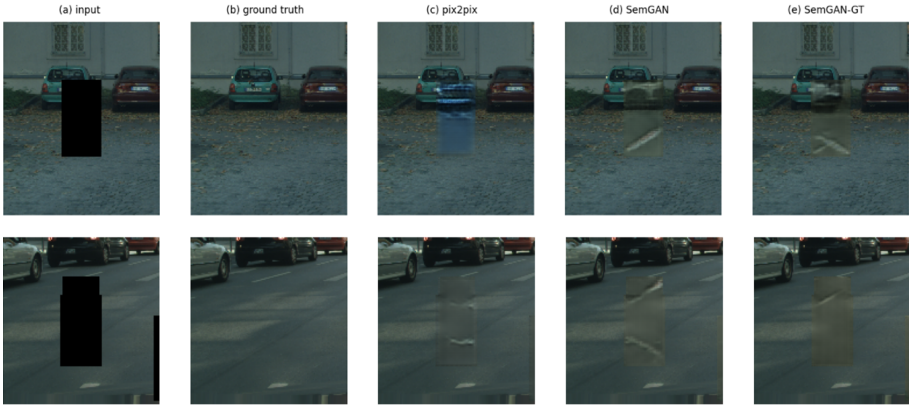


Figure 4.5: Ground truth images (a), compared with inpainted images. This figure visualizes the color bleeding problem.

Generally, models have a hard time inpainting objects with complex structures. This is another challenge that all models struggle with. We have visualized some examples of complex structures and how the different models deal with these in Figure 4.6. In the top row, an image of a street with a tile pattern is depicted, with thin edges dividing the tiles. Matching the pattern of the tiles proves to be a challenging task as all of the models fail to match the pattern of the tiles. Instead the models generate blurry patches that only match the color distribution of the street, lacking the necessary structure. The second row shows a building which has a complex structure, as it features multiple windows, some of which are partially closed. Although a large part of the building is masked, there is still a lot of visual information about the structure of the building. Pix2pix completely fails to model the complex structure. The inpainted patch contains mostly noise and visual artifacts, resulting in an unrealistic appearance. SemGAN and SemGAN-GT also produce noise and some visual artifacts, but better match the color distribution. The third row displays a store that is masked for the most part, hence there is not much visual information about the contents of the store. Again, Pix2pix generates a lot of noise and visual artifacts. SemGAN and SemGAN-GT generate considerably less noise, but at the cost of generating a blurry patch that matches the color of the concrete structure of the building. Both models still produce visual artifacts. The bottom row's image (a) presents a partially masked bicycle, where the front wheel is entirely covered, and the back wheel is partially masked. Both pix2pix and SemGAN fail to model the wheels of the bicycle and instead generate sections of the road. However, SemGAN-GT succeeds in modelling the bicycle as it utilizes the ground truth segmentation map, which includes the structure of the bike.

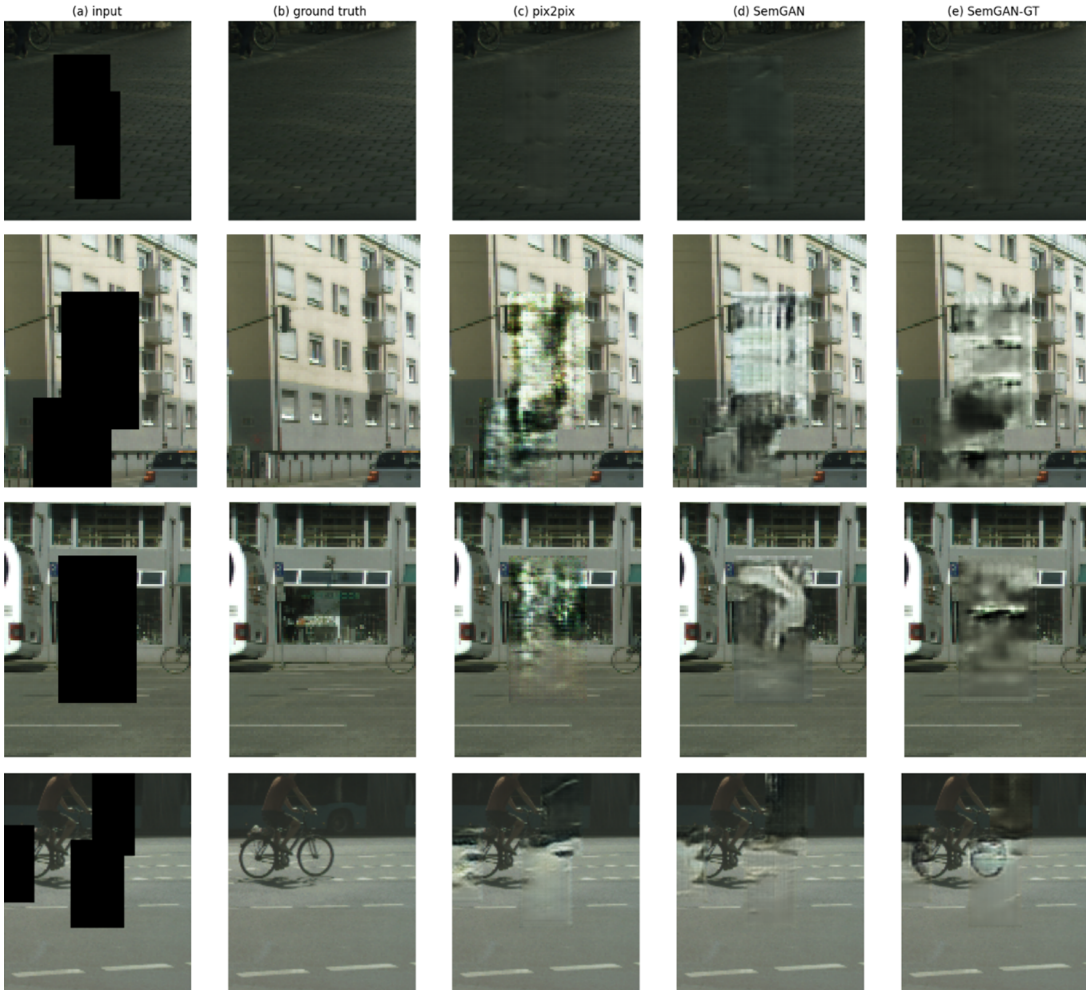


Figure 4.6: Ground truth images (b), compared with inpainted images. This figure visualizes the problem with complex structures.

Label	Object	Label	Object	Label	Object
6	Ground	15	Bridge	25	Rider
7	Road	16	Tunnel	26	Car
8	Sidewalk	17	Pole	27	Truck
9	Parking	19	Traffic light	28	Bus
10	Rail track	20	Traffic sign	31	Train
11	Building	21	Vegetation	32	Motorcycle
12	Wall	22	Terrain	33	Bicycle
13	Fence	23	Sky		
14	Guard rail	24	Person		

Table 4.3: Labels and their corresponding objects as specified by Cityscapes (Cordts et al., 2016). We have only included the labels that are present in the confusion matrix.

4.2 Semantic information

The focus of this section is to analyze the outcomes of the semantic inpainting model, which will be assessed through the utilization of a confusion matrix. Additionally we will analyse the impact of misclassifications in the semantic map on the inpainting result of an image.

4.2.1 Semantic inpainting

In this section we will discuss the results of the semantic inpainting model, the model that outputs the inpainted semantic map used as a prior for SemGAN.

The IoU score of the semantic inpainting model is 0.47 for the validation set. Figure 4.7 shows the confusion matrix, where the colors represent the number of times that the ground truth and the predicted labels overlap. We have chosen a logarithmic scale as this makes the confusion matrix easier to interpret. The diagonal line in the confusion matrix represents the instances where the model predicted the correct semantic label. The values along this line indicate the number of correct predictions made by the model for each class. The higher the value along this diagonal line, the better the model performed in predicting that class. The values off the diagonal line represent the instances where the model made incorrect predictions. These values indicate the number of instances where the model predicted a different semantic label than the ground truth. If a class has a high number of incorrect predictions, it indicates that the model struggles with predicting that class. The low IoU score and the absence of a clear diagonal pattern in the confusion matrix indicates that the semantic inpainting model is ineffective at inpainting the masked semantic segmentation map. The unaltered confusion matrix without logarithmic scale can be found in Appendix A.

When we closer examine the confusion matrix we notice some remarkable patterns. The first thing we notice is that the model entirely excludes some labels from its predictions. The excluded labels can be recognized by an empty vertical column, these labels are ground, rail track, bridge, traffic light, traffic sign, rider, truck, bus, train and motorcycle (see Table 4.3 for the corresponding labels). This is unexpected given that these labels do occur in the dataset, as we can see from the corresponding rows. The ground an rail track are mostly confused with the road and sidewalk labels, where ground is specified as all other forms of horizontal terrain that do not match any of the other labels. The proximity of ground and rail track to the road or sidewalk labels in the training data appears to be a primary contributing factor to the confusion between these classes. In addition, rail track and ground are both horizontal ground-level structures, which might also contribute to the confusion with roads and sidewalks.

The labels for truck, bus, train, and motorcycle are frequently misclassified as belonging to the car class, which is one of the most common categories in the dataset. This is likely due to the fact that these classes all represent vehicles and therefore share features with respect to their shape and position within the images. In addition, these vehicles co-occur with the label road most of the time.

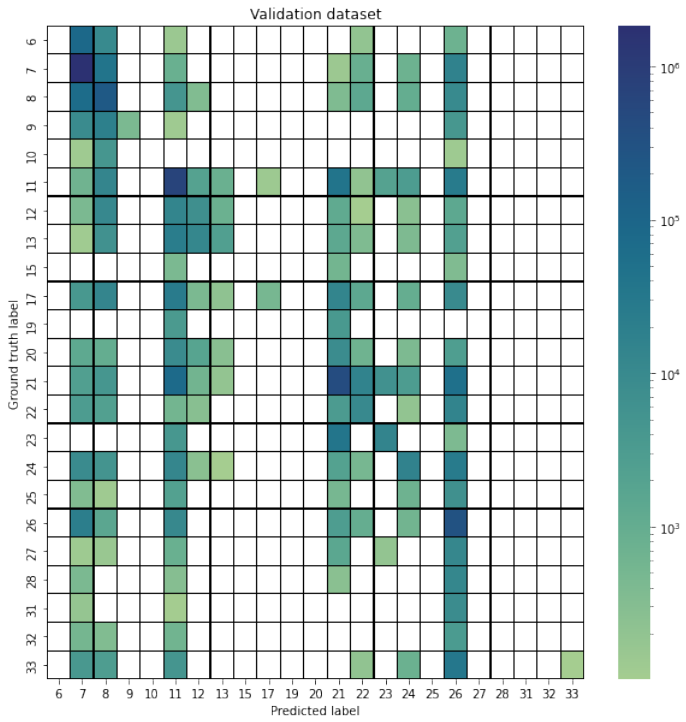


Figure 4.7: Confusion matrix of the semantic inpainting model calculated for the validation data. The y-axis represents the ground truth labels, while the x-axis represents the predicted labels. Labeled pixels that co-occur less than 100 times are set to 0. All rows and corresponding columns that contained only zeros are removed.

Traffic lights and signs are mostly confused with building and vegetation. For the evaluation of the issue concerning these objects, it is important to note that, the labeling policy distinguishes the pole attached to either of these objects as a separate class. Hence, the edge problem does not contribute to the misclassification of these objects. Both signs and lights are salient objects and tend to have a background that predominantly belongs to either of the two classes with which they are frequently confused.

Another noteworthy aspect is that the majority of predicted labels are either road, sidewalk, building, vegetation, and car. These classes are also most common in the Cityscapes dataset. Therefore, it seems that the model minimizes its risk of errors by consistently predicting the labels that occur most often.

4.2.2 Impact of semantic misclassifications

Previously, we have mentioned the issue that errors in the inpainted semantic segmentation map may have a cascading effect on the inpainting of the corresponding image. Both SemGAN and SemGAN-GT rely on semantic information to guide the inpainting process. The former utilizes inpainted semantic segmentation maps, while the latter uses ground truth semantic information as a prior. When the inpainted semantic maps contain errors or inaccuracies, those errors are propagated to the inpainted images. If the semantic segmentation map is incorrect or incomplete, SemGAN may generate incorrect regions in the inpainted image. This occurs because the model tries to fit the masked patches to the erroneous semantic segmentation map, leading to errors or artifacts in the inpainted patches. We have visualized some of these erroneous semantic maps and their corresponding inpainted image in Figure 4.8.

In the upper row we see that the semantic inpainting model uses incorrect classes to inpaint the masked patch. The patch mainly consists of the classes: Road (light purple), sidewalk (pink), car (blue) and vegetation (green), while the correct classes are: Ground (purple) and building (grey). Although the patch in (f) lacks the visual features of any of the misclassified classes, it does contain significant noise. SemGAN fails to effectively inpaint the door and wall. Conversely, (g) achieves better results in replicating the missing information. In the second row both SemGAN and SemGAN-GT succeed

in inpainting the road effectively. SemGAN-GT also succeeds in completing the missing and bicycle. However, due to the presence of errors in the predicted semantic segmentation map (b), SemGAN (f) displays visual artifacts at the locations where these errors occur in the semantic information. The third row demonstrates that when the ground truth semantic segmentation map is used, the objects in the image are clearly defined, resulting in realistic image inpainting in (g). Both the road and bike are accurately completed. The predicted semantic information (b) is imperfect as parts of the bicycle and road are classified as "car". It might be expected that misclassification would cause a significant portion of the car to be generated in (f). However, once again, misclassifications mostly lead to noise and visual artifacts in the images inpainted by SemGAN. This can also be seen in the bottom row, where most of the patch is misclassified as "car". As a result, the inpainted patch becomes noisy, whereas SemGAN-GT successfully restores the patch, even managing to reconstruct the human silhouette in (g).

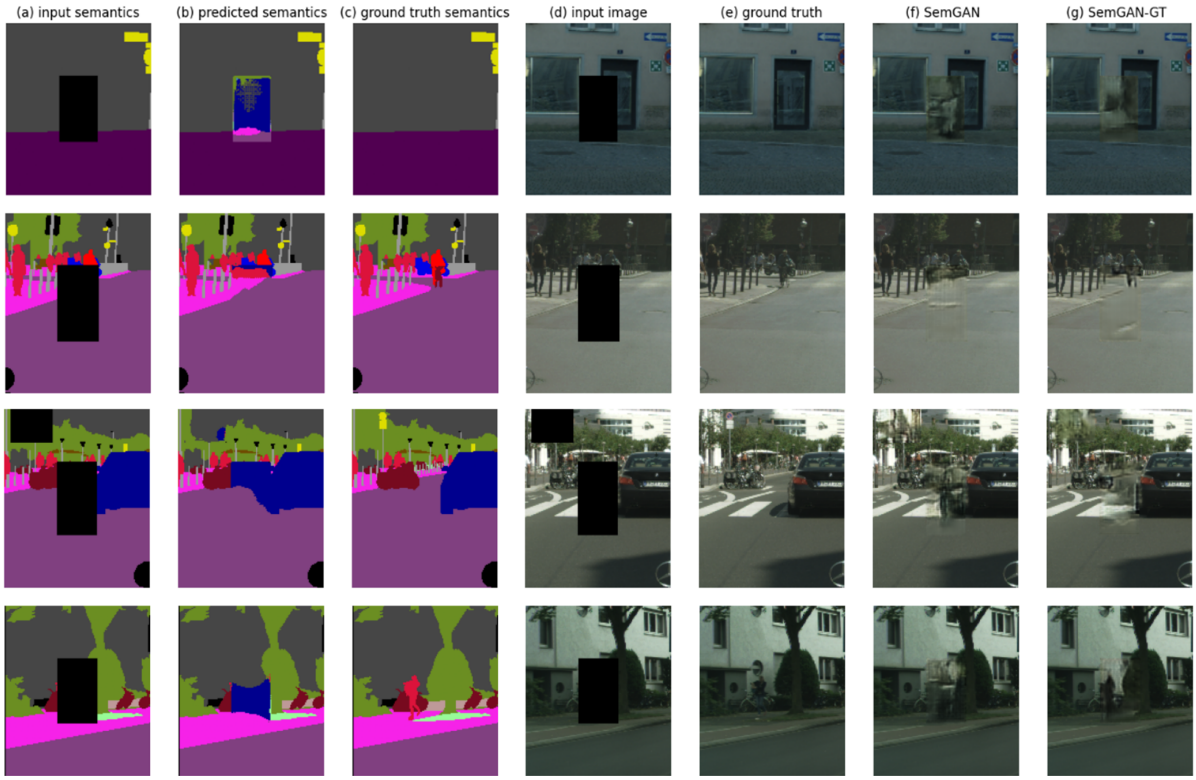


Figure 4.8: The inpainted semantic segmentation map (b) inpainted by our semantic inpainting model, where (a) is the input and (c) is the ground truth map. The impact of errors in (b) are visualized in the patches inpainted by SemGAN (f), where (b) and (d) are the combined input to this model. The ground truth (e) and the patches inpainted by SemGAN-GT (g) are included for comparison purposes.

4.3 Person inpainting

In this section we will evaluate the results of our experiments on person inpainting, and compare these results across models.

4.3.1 Fine-tuning

Fine-tuning for the task of person inpainting is required as there is a difference in underlying distribution between the rectangular masked dataset and the person masked dataset. Therefore, untuned models, i.e. models that are only trained on the rectangle inpainting task, are not able to effectively inpaint person shaped masks. Figure 4.9 shows a selection of inpainting results on the person mask validation set. The results are generated by the different untuned models. The input image in the upper row contains a smaller person mask where there are people present in the background. The inpainted patches created by pix2pix are most salient, for all rows, as they exhibit colors that significantly deviate from their surrounding background. This problem is less noticeable in the images inpainted by SemGAN and SemGAN-GT. Image *c* in the top row reports a worse PSNR and L1 score, while it also presents a better SSIM score than *d*. The same holds for the middle and bottom row. These conflicting scores are a recurring phenomenon in our inpainting tasks, and show the necessity of combining multiple metrics.

While these models are capable of inpainting masked images to a certain degree, the inpainted regions often contain considerable noise and artifacts, even when dealing with images that have smaller masks. The results from Table 4.4 confirm that the issue is not specific to the selected images, but to all images in the validation set. The results reflect that the untuned models perform worse than their fine-tuned counterparts, especially for L1 and PSNR. SSIM shows only small differences in performance, however, as explained in 3.2.4 this is due to the image-wise comparison.

	Model	L1 ↓	SSIM ↑	PSNR ↑
Untuned	pix2pix	4302 ± 3979	0.9591 ± 0.0315	30.68 ± 4.23
	SemGAN*	3777 ± 3298	0.9613 ± 0.0288	31.41 ± 4.04
	SemGAN-GT	**3532** ± 3055	**0.9631** ± 0.0267	**31.86** ± 3.84
Fine-tuned	pix2pix	3255 ± 2563	0.9652 ± 0.0227	32.27 ± 4.10
	SemGAN*	3174 ± 2617	0.9657 ± 0.0253	32.74 ± 4.13
	SemGAN-GT	**2887** ± 2601	**0.9682** ± 0.0239	**33.42** ± 4.34

Table 4.4: L1, SSIM and PSNR outcomes for the fine-tune validation dataset across all models for the person inpainting tasks for the untuned and fine-tuned models. This validation dataset contains 98 images. The metrics are averages from five independent training runs. *The IoU for the untuned SemGAN model is 0.49. The IoU for the fine-tuned SemGAN model is 0.27.

Figure 4.9: The performance of our untuned models trained on the task of inpainting rectangular masks was evaluated using input images from the validation training dataset. The L1, PSNR, and SSIM scores, are reported above the corresponding inpainted images.

The results of the models fine-tuned on the person masked dataset are displayed in Figure 4.10. When we visually compare these to the result in Figure 4.9, we see that the inpainted patches are much smoother and more closely resemble the colors and texture of their surroundings. Although the inpainted patches are still vaguely visible, this displays the effectiveness of the fine-tuning procedure. Especially for the images located in the top row, where some effort is required to locate the inpainted patches. Upon making a comparison between the inpainted patches by the fine-tuned models, based on their metrics, we see that each row has a different best performing model. In the top row pix2pix excels, in the middle row SemGAN-GT is best and SemGAN performs best in the bottom row. Although the metrics suggest that different models perform best on various evaluation criteria, a visual inspection of the results reveals that SemGAN-GT produces the most visually appealing output. This is evident by the smoothness of the inpainted patches, as well as the absence of significant visual artifacts or color distortions. This is as opposed to SemGAN, which generates patches that contain a lot more visual artifacts and color distortions. The patches inpainted by the pix2pix model present less color distortions but exhibit noticeable block artifacts.

Despite being fine-tuned, the models encounter the same challenges as their untuned counterparts for the rectangle inpainting task. The edge problem still persists, which becomes apparent when we look at the bottom row. None of the three models are able to successfully complete the tram tracks or the white pavement markings situated on the left side of the tram tracks. Another common problem that remains unsolved is the visibility of inpainted patches due to color bleeding. This is especially noticeable in image *d* in the middle row, where all patches contain shades of orange, that not follow the color distribution of the image.

Figure 4.10: Person inpainting results from the different models that are trained on the rectangle inpainting task and fine-tuned on the person mask training dataset. Input images are from the validation dataset. The L1, PSNR, and SSIM scores, are reported above the corresponding inpainted images.

4.3.2 Person anonymization

This section will present the results of the full person anonymization process. However, instead of evaluating the degree of realism of the inpainted patches, we will qualitatively assess the level of anonymity of the inpainted images. The validation dataset consists of 402 images, where each image contains at least one person. The inpainted images in this section are generated by pix2pix and SemGAN. SemGAN-GT cannot be used for this task, as the ground truth semantic segmentation maps used by SemGAN-GT contain labeled persons, which is information that must be removed for the anonymization task.

The results of the anonymization process are shown in Figure 4.11. The ground truth images are the input for the person removal, which is done using the ground truth semantic segmentation maps. The pixels labeled are replaced with constant values, resulting in the input images in column *a*. Pix2pix takes *a* as input, while SemGAN also uses *c*, the inpainted semantic segmentation map, as a prior. In both column *d* and *e* parts of silhouettes are still visible. This is most apparent in the second and fifth row, and is mainly caused by color bleeding. In the fifth row this happens because most of the silhouettes have the same color as the pavement, both models are unable to match the color distribution of the building. The silhouettes in the second row are blurry and mostly match the color distribution of the road. However,

Figure 4.11: Anonymization results on the Cityscapes validation dataset.

silhouettes that are positioned in front of building are therefore easily visible as they differ in color.

Despite some silhouettes being partly visible due to issues such as color bleeding, noise, or visual artifacts, it appears highly unlikely that the identity of the persons in the image can be retrieved, even using the geo-location and the silhouette of the person. This is mainly because of the absence of visual personal identifiers, biometric and soft biometric identifiers are removed as they belong to the physical features of a person. Additionally, non-biometric identifiers are excluded due to the labeling policy of Cityscapes, which assigns these identifiers to the person category.

4.4 Data reduction

In this section we examine how reducing the amount of training data affects the performance of the pix2pix model. A common method to improve performance of a deep learning model is to increase the number of training samples. Since we already use the entire finely annotated training dataset, this is not an option. However, we can hypothesize about the effect of adding training samples by evaluating a model's performance as we train it on increasingly larger proportions of the available training dataset.

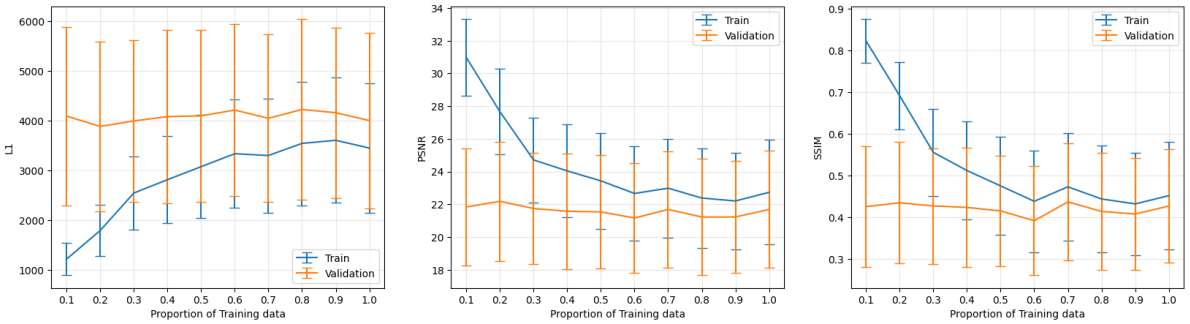


Figure 4.12: Pix2pix performance on the data reduction experiment.

The displayed results in Figure 4.12 exhibit the outcome of a data reduction experiment where we evaluate the pix2pix model after training it on various proportions of the training dataset for 30.000 steps. The model's performance was assessed through 5 independent training sessions for each proportion, and the outcome is presented as the average and standard deviation of these independent training runs. The results are rather unexpected, as more data usually leads to better performance.

When we focus on the performance of the model on the training dataset, we observe an increase in L1 and a decrease in SSIM and PSNR which means that the performance deteriorates with an increase in the proportion of the training dataset. Additionally, we notice an increase in variability. The high training scores, in combination with low variability and a significant gap between the model's performance on the training and validation datasets, clearly indicate overfitting. The signs of overfitting become less pronounced as we use larger proportions to train on.

The unexpected happens in the performance on the validation dataset. Our hypothesis was that the performance on the validation dataset would also improve as we increased the amount of samples in the training dataset. However, there seems no clear pattern that indicates the improvement in performance for all three metrics. To elaborate on this, a minor reduction in performance is observed for all three metrics between 0.2 and 0.6. While the model attains its peak L1 and PSNR scores at 0.2, the highest SSIM score is attained at 0.7. A slight performance reduction is observed again from 0.7 to 0.9, followed by an increase at 1.0. In all three charts, the training and validation data exhibit a similar trend beyond 0.6, indicating that the training scores may be merely an offset of the validation scores. Although all three metrics measure different aspects of an image, we see that they mostly follow the same pattern.

There are some fluctuations in scores, indicating that the performance of the model may not be entirely stable for these proportions of the training dataset. In addition, we notice the relatively large standard deviations for the various metrics, which suggest that the model's performance may be highly inconsistent for various proportions of training data. The instability of the process might be a contribution to the unexpected results or our experiment, this will be part of our discussion.

Visual inspection allows us to qualitatively evaluate the images generated by pix2pix, trained on different proportions of the training dataset. Figure 4.13 shows how the different models inpaint the incomplete input image. Specifically, the input image consists of four different masked patches: the first mask obscures a portion of the car and the road, the second mask covers mostly vegetation, the third mask covers the road and a tram rail, and the fourth masks a traffic light and a section of a building.

The first mask shows that all models can generate somewhat realistic patches, but there are still some noticeable visual artifacts and noise present, especially in the 0.1 and 0.2 proportions of training data. We also see color bleeding in 0.3 and 0.9. At first glance, it is difficult to distinguish the inpainted patch

Figure 4.13: Input image, ground truth and images generated by pix2pix after training on different proportions of the training dataset.

for the second mask as the generated patches appear to be quite realistic. The third and fourth inpainted patches exhibit varying degrees of visual imperfections such as noise, artifacts, and color bleeding, which are more prominent in certain proportions than others. As a result, the generated patches may be easier or more difficult to spot.

As with the quantitative results of our experiment, it is difficult to observe a clear pattern in the inpainted results. Overall, it seems that the inpainted patches become more difficult to spot for higher proportions, but they still exhibit visual artifacts and color bleeding.

4.5 Discussion

In this section, we provide a concise overview of the outcomes of our research and emphasize the key findings, limitations, and potential directions for future work. We discuss our proposed method for inpainting along with the outcomes of their respective studies. In addition we present our key findings of our studies. We also describe the limitations of our research. We conclude this chapter with the broader implications of our work and future research directions.

We proposed two conditional Generative Adversarial Network (cGAN) models as an alternative to existing anonymization techniques outlined in Section 2.3. Our objective was to achieve a higher level of anonymity, and to this end, we developed two models based on Isola et al. (2017)'s image translation cGAN (pix2pix). SemGAN employs inpainted semantic segmentation maps, produced by our semantic inpainting model, as a prior, whereas SemGAN-GT uses ground truth semantic segmentation maps. The quantitative results for the models are summarized in Tables 4.1 and 4.2.

The results indicate that the incorporation of semantic segmentation maps positively influences the quality of the inpainted patches, in terms of realism, according to the L1, peak signal-to-noise ratio (PSNR) and structural similarity index measure (SSIM) metrics, as well as visual inspection. SemGAN-GT outperforms SemGAN, as the error in the imperfect semantic segmentation maps propagates through to the inpainted RGB image. A major cause of errors in the semantic inpainting model comes from the fact that the model tends to predict the most common labels consistently. This strategy seems to minimize the risk of errors, but it can result in inaccuracies, especially in regions with mixed scenes or rare object labels. The anonymization demands some fine-tuning as the complex shape of human silhouettes results in a difference in underlying distribution between the rectangular masked dataset and the person masked dataset. After fine-tuning we see an increase in both qualitative and quantitative results. A common method to further improve the performance of the model would be to increase the number of samples in the training dataset. However, this is not necessarily the case, as indicated by our data reduction experiment. We did not observe any significant improvement in any of the three metrics as we increased the proportion of samples used to train the model.

The results for the rectangle inpainting task indicate that our SemGAN and SemGAN-GT outperform the pix2pix model significantly in terms of realism, according to the L1, PSNR and SSIM. This improvement is mainly attributed to the addition of semantic segmentation maps as a prior, to guide the inpainting process. In our qualitative analysis we have visualized the inpainting results of the models, in which the superiority of semantic aware inpainting models is demonstrated. Although the patches inpainted by SemGAN occasionally contain an increased level of noise or visual artifacts, likely caused by errors in the inpainted semantic map, the results are visually more appealing than the patches inpainted by pix2pix. The impact of the inclusion of semantic information is most clearly demonstrated by SemGAN-GT, which generates inpainted patches that are qualitatively superior to inpainting results of the other models. The model displays increased performance in inpainting objects that are partly masked, based on visual inspection. Especially in mixed scene inpainting where a masked patch contains various object classes. Notably, SemGAN-GT demonstrates its ability to generate patches where the different object classes are clearly distinguishable from one another by precisely defining the borders between these objects. The quantitative results in Table 4.2 also demonstrate the superiority of SemGAN-GT in mixed scene inpainting. Here we made the assumption that the difficulty of inpainting a certain patch can be based on the number of semantic classes inside that patch. Based on the decrease in performance as the difficulty increases, we could argue that the number of semantic classes in a patch is indeed a good proxy for difficulty. Generally, objects from different semantic classes have different texture. However, some semantic classes have a great variety in textures for their objects, such as buildings and vegetation. Although this is a good proxy for difficulty, as we see the performance decrease as the difficulty increases, difficulty is also determined by the different structures inside a patch. Furthermore, SemGAN-GT demonstrates a greater resistance to the edge problem compared to other models, as it has the ability to complete edges when they are present in the semantic segmentation map, leading to more accurate and consistent inpainting results. To conclude, visual inspection and the quantitative results in Figure 4.2 indicates that semantic information also increases the ability to inpaint homogeneous scenes, where there is only one object class present in the patch.

The performance gap between SemGAN and SemGAN-GT can be attributed to the use of imperfect semantic segmentation maps as input to the former model, while the latter uses perfect ground truth

information. The semantic inpainting model performs poor, indicated by an IoU of 0.47 on the validation dataset and the lack of a clear diagonal pattern in the confusion matrix. The model models predicts the majority of the labels as either road, sidewalk, building, vegetation, and car, which are the most common classes in Cityscapes. A possible solution could be the substitution of the current loss function, categorical cross-entropy, with another function such as weighted cross-entropy loss or Dice loss (Sudre et al., 2017). The former allows the assignment of weights to each class based on its frequency in the training dataset. So, rare classes have a higher weight and more common classes have a lower weight. The Dice loss measures the overlap between the inpainted and ground truth segmentation maps, instead of the dissimilarity like categorical cross-entropy does. The Dice loss is effective in its ability to handle unbalanced datasets by imposing a higher penalty on the GAN for making mispredictions on the minority classes. The limitations of including semantic information as a prior are mainly attributed to errors in the corresponding map, which is demonstrated in Section 4.2.2, where we visualize and discuss how errors in the semantic information can propagate to the inpainted image. An incorrectly predicted map mainly leads to noise and visual artifacts in the inpainted image. One might expect that the model would inpaint textures solely based on the semantic information. As an example, if a semantic map incorrectly classifies a wall as vegetation, it could result in leaves appearing in the inpainted image patch. However, this is not observed in the results, as the model primarily relies on information from the input image. Semantic information is mainly employed to guide the inpainting process and determine the location where textures from the surrounding image should be incorporated into the patch. The poor performance of the semantic inpainting model propagates to SemGAN, as it performs worse than SemGAN-GT. The model's inadequate performance on the validation dataset with an IoU of 0.47 is the main reason for the discrepancy between the two semantic models. However, another cause for the poor performance of the semantic inpainting model could be attributed to the simplicity of its architecture, particularly with regards to the generator. This also concerns both SemGAN and SemGAN-GT and will therefore be further discussed below.

Above we have answered our first research sub-question: "How does the inclusion of semantic segmentation data influence the quality of the generated images?". To summarize, semantic segmentation maps guide the inpainting process by offering additional information about objects, their boundaries and structure of the image. As evidenced by the metrics and visual evaluation, incorporating this information as a prior leads to higher quality inpainted patches than would be possible without it. Despite the fact that imperfect semantic information can introduce noise and visual artifacts, even with such imperfections, the results obtained are superior to those obtained with pix2pix, as indicated by the metrics. These results build on existing evidence of the positive effect of adding priors to GANs, as demonstrated by Nazeri et al. (2019). In their work, they utilize edge information as a prior to enhance the quality of inpainting results.

To answer our second research sub-question: "How does the reduction of training data influence the performance of the model?" we have conducted an experiment to evaluate how the reduction of data impacted the performance of pix2pix. A common and straightforward method in deep learning to increase the accuracy of the model is to increase the number of training samples. A larger dataset allows the model to learn from a more diverse distribution and therefore generalize better to unseen data. Gradually increasing the number of training samples is anticipated to improve performance, but there may be diminishing returns as the dataset becomes larger. Eventually we would reach a saturation point beyond which further additions to the dataset will not enhance the accuracy anymore. However, since GANs are notoriously challenging to train, this approach may not be as effective. We evaluated the performance of our GAN as we increased the proportion of the training dataset. This ranges from a proportion of 0.1 to 1.0 which is the entire Cityscapes training dataset with 2975 samples. Contrary to the hypothesized association, the performance on the validation set does not show any clear signs of improvement across all three metrics.

The unexpected results could be explained by some of the limitations in the experiment, such as the stochastic nature of the process, difficulties of GAN training and the limitations of the metrics themselves. The stochastic nature could make some subsets better to train a model on than other subsets. This could be a result of the randomness in the mask location and size. Larger masks on locations which have a lot of different textures and colours are more difficult to inpaint than smaller masks with simple textures and fewer different colours. Since the masks vary for each image in both the training and the validation set, it could be possible that in some of the subsets the masks are easier to predict. This could be evident by the large variability in all metrics, meaning that there is a large difference in difficulty between

images. The difficulties of training GANs could also be a limiting factor in this experiment. Balancing the generator and discriminator is a difficult process. Since we train the generator and discriminator simultaneously, changes to one model also affect the other model. A common problem is a diminishing gradient which is caused by a successful discriminator. The successful discriminator classifies each image correctly which leads to a vanishing gradient for the generator, inhibiting the generator to learn. The metrics themselves also are a possible limitation. Although these are the most common metrics used in literature to analyse GAN performance, there also is a disagreement about the most suitable. The L1 metric calculates the pixel-wise distances between the inpainted pixels and the ground truth pixels. Hence, L1 mainly measures colour differences between inpainted and ground truth regions. However, in order to be perceptually similar, we should care more about structure similarity. That is, the inpainted region should look realistic. PSNR describes the ratio between the maximum value of a pixel and the noise (MSE) affecting the pixel. So both L1 and PSNR measure the noise in an image. The final metric we use is SSIM, which as the name suggests is a better metric for analysing the structural similarity. SSIM adopts the Human Visual System (HVS) that takes into consideration the luminance, contrast and structure in an image. In contrast to L1 and PSNR, which sum the error, SSIM uses a combination of different comparison functions to form the similarity measure. Although these metric have a varying approach in measuring the differences between two images, we see a strong correlation between these metrics in the graphs in Figure 4.12. Finally, the architecture could be a limiting factor. Our current architecture could be too simple to capture the complex patterns and details that are present in the images. If the generator is too simple or too shallow, it may not be able to capture the intricate patterns and features present in images, which results in less realistic inpainted images. The generator used by pix2pix is based on U-Net (Ronneberger et al., 2015). However, many more recent inpainting models have adopted the architecture proposed by Iizuka et al. (2017), which uses dilated convolutions to capture complex spatial relations between different regions in an image. Nonetheless, this architecture was not appropriate as a foundation for our model, as it resulted in considerably longer training and inference times. Another way in which the GAN might be too simple is if the discriminator is not able to make an accurate distinction between real and fake images. A simple discriminator might not be able to capture the small differences between real and fake images, which results in the generator network not receiving sufficient feedback to generate realistic images. These shortcomings might also result in poor performance of the semantic inpainting network.

Fine-tuning is required to fit the models to the data used for person inpainting. Although the models trained on the rectangular inpainting task are capable of inpainting rectangular patches, the difference in distribution between the rectangular masked dataset and the dataset containing person-shaped silhouettes hinders the transfer of this capability to the inpainting of person-shaped silhouettes. The models fail to effectively inpaint these patches as can be seen in Table 4.4, where we see a large gap between untuned and fine-tune performance. Figure 4.9 allows for a visual inspection of the shortcomings of the untuned models on the person inpainting task. The fine-tuning process is a challenging task, as it involves numerous hyperparameters. The number of epochs, learning rate, batch size, Λ, optimizer and learning rate decay all impact the learning process differently. Since the weights are already partially optimized, we aim to fine-tune the model without making large modifications to these weights. Due to constraints in time and computing resources, we were unable to perform an extensive amount of hyperparameter tuning during the fine-tuning process, in comparison to the regular training process. Therefore, with more extensive hyperparameter tuning we expect the performance on the different metrics to increase.

Our results and the discussion above allows us to answer our research question: “Can a Generative Adversarial Network anonymize street level images, in which human individuals are removed, while the intelligibility of the images is retained?”. Based on the results of pix2pix and SemGAN we can conclude that GANs are able to anonymize street level images. The intelligibility of the images is retained to some degree, however there is still room for improvement. The performance difference between SemGAN and SemGAN-GT suggests that enhancing the accuracy of the inpainted semantic segmentation map can result in a better quality of the inpainted RGB image. Simply increasing the training dataset will not directly lead to improved performance.

4.5.1 Limitations and future work

In this section, we describe the limitations of our work, which arise from various sources such as the subjective nature of the task, the approach we take towards anonymization, and the use of generative

models and data.

Person removal method

For the anonymization process we have used the ground truth semantic segmentation maps to remove all the persons from the images, while leaving the immediate surroundings intact. These maps are annotated by a team of experts, and therefore contain no errors. In addition, all persons are (almost) perfectly annotated, which minimizes both the loss of valuable background information and the possibility of personal identifiable information still being present in the images. Another advantage is that, besides the annotation of the persons themselves, the experts also included non-biometric identifiers such as hats and bags However, developing such a semantic segmentation model that perfectly classifies each pixel is nearly impossible. The current state-of-the-art semantic segmentation model is ViT-Adapter-L (Z. Chen et al., 2023), which yields a mean IoU of 85.8 on the Cityscapes validation dataset. From this we can conclude that our inpainting model is not the limiting factor in the anonymization process, but the segmentation model used for the person removal is. The implementation of our method on another dataset that does not include semantic segmentation data, requires such a state-of-the-art segmentation model for two reasons: 1) The number of unlabeled persons should be minimized, and 2) objects should be labeled unambiguously to ensure that SemGAN is able to learn the correct relation between objects and labels.

Generative Adversarial Networks

While the use of GANs for image generation and more specifically image inpainting has shown promising results, there are several limitations to this approach. Firstly, training GANs is challenging due to the adversarial nature of the generator and discriminator, which are interdependent and have opposing objectives. Secondly, the choice of GAN architecture can greatly affect the quality of the generated images. Due to computational constraints, we opted for a relatively simple GAN architecture, while more recent studies on image inpainting utilize a more complex model based on the approach proposed by Iizuka et al. (2017). However, our research is still valid as it provides insights into the limitations and challenges of using GANs for image inpainting tasks. Additionally, our results demonstrate the potential of adding semantic segmentation maps as a prior to GANs for inpainting. Thirdly, challenges arose in tuning hyperparameters for each setup, particularly for the fine-tuning of our SemGAN model, as it consists of two separate GANs. We anticipate that further extensive hyperparameter tuning would enhance the performance of our models. Furthermore, mode collapse is a well-known issue with GANs, and it was a significant limitation of our models and the baseline models.

Dataset

The stochastic nature of our rectangle masking algorithm makes it more difficult to compare between images. The size and location of the masks vary, and there is frequent overlap, resulting in a smaller masked area of the image, which can make inpainting easier. This might be one of the main contributions of the high variability in quantitative results. Most of the finely annotated pixels in the Cityscapes dataset belong to road, sidewalk, building, car, and vegetation classes. This makes it more dif

Metrics

Although L1, PSNR and SSIM are popular metrics used in research to evaluate image inpainting quality, they have limitations that should be considered when interpreting the results. L1 and PSNR are based on comparing pixel values of inpainted patches with their corresponding ground truths. These metrics can provide a quantitative measurement of how similar the inpainted patch is to its ground truth, they do not capture semantic content or realism of the inpainted patches. We have come across images with a low L1 and high PSNR scores, but still look unrealistic upon visual inspection. SSIM appeared to be better at evaluating the images, as high scores correlated more with realistic looking images, and vice versa. Besides the pixel-wise similarity, SSIM also takes into account the structural and perceptual similarity,

which makes it a more suitable metric for image inpainting. All metrics have strengths and weaknesses, a combination of metrics, such as PSNR, L1, and SSIM, can provide a more robust evaluation of the quality of the inpainted images, as it considers different aspects of the image quality.

Future work

In comparison to the anonymization works summarized in 2.3, we proposed a new method of anonymization which protects the privacy of the individuals in the image, but also retains the intelligibility and overall realism in the image. While our proposed method outperforms the standard pix2pix model, there is still room for improvement in terms of the realism and overall appearance of the inpainted patches. Therefore, we recommend that future work should concentrate on enhancing the quality of the inpainted regions, aiming to achieve more convincing and visually appealing results. A good starting point would be to implement a more complex architecture, introduce an additional loss function such as perceptual loss (Johnson et al., 2016) or texture loss (C. Yang et al., 2016), or utilize additional prior such as edge maps (Nazeri et al., 2019). In terms of the evaluation, additional metrics that are more capable of analysing the quality of images should be investigated.

Chapter 5

Conclusion

The aim of our research was to tackle the issue of safeguarding privacy in image datasets. We accomplished this by creating a technique that can anonymize individuals while preserving the authenticity and natural appearance of the images. We based our approach on the conditional Generative Adversarial Network proposed by Isola et al. (2017). This led us to propose two models, namely SemGAN and SemGAN-GT, specifically designed for the inpainting of street-level images in locations where identifiable people have been removed. Our method utilizes the available information from both the incomplete image, where individuals are absent, and the semantic segmentation maps to achieve the desired outcome. This is accomplished by combining the information from both sources. SemGAN and SemGAN-GT differ in their architecture. SemGAN is comprised of two models: a semantic inpainting model and an RGB image inpainting model. The latter employs the inpainted semantic segmentation map generated by the former as a prior for the inpainting process. In contrast, SemGAN-GT uses ground truth semantic segmentation maps as a prior for the inpainting task.

To assess the effectiveness of our models, we employed standard evaluation metrics commonly used in research, including L1, PSNR, and SSIM. We then compared the performance of our models against that of our base model, pix2pix. Supported by visual evaluation, our experiments reveal that the use of semantic segmentation data improves both the quantitative as well as the qualitative results. Our analysis revealed a notable gap in the performance between SemGAN and SemGAN-GT, likely caused by the propagating error from the semantic segmentation maps to the RGB inpainting process. Despite SemGAN's inferior performance compared to SemGAN-GT, our findings indicate that leveraging semantic segmentation data enhances the overall quality of the inpainted patches, as compared to pix2pix. The misclassifications the semantic segmentation map for SemGAN mostly cause noise and visual artifacts. The observed performance gap between these two models suggests that enhancing the semantic inpainting process could lead to improved results by SemGAN.

We have trained and evaluated our models on a rectangle inpainting task. This allows for a consistent comparison, as there are no significant differences in the size of the masks between the images. However, this task does not transfer well to the inpainting of person-shaped masks, as there is a difference in distribution between the rectangular masked dataset and the dataset containing person-shaped masks. By fine-tuning the model specifically for the task of person-shaped mask inpainting, we demonstrate that our approach is capable of effectively removing sensitive information from street-level images while preserving the visual quality and integrity of the scene.

The proposed method has potential applications in the domain of privacy-preserving data sharing and analysis, particularly in the context of urban planning, transportation, and public safety. The ability to anonymize street-level data while maintaining the visual quality and contextual information is important for the protection of individual privacy and the promotion of ethical data usage.

Beyond the model improvements and use of additional evaluation metrics, there are several directions of future work to improve the generation of realistic anonymized images. Firstly, since GANs are difficult to train a potential avenue is to explore the use of diffusion models for inpainting tasks. Diffusion models do not suffer from mode collapse and are therefore able to generate a greater diversity of samples. In

addition, diffusion models are known to be more stable and easier to train than GANs, requiring fewer hyperparameters and less tuning. Furthermore, diffusion models have been shown to perform well in image inpainting tasks, particularly in maintaining the structure of the inpainted regions.

Another potential avenue is to explore the application of our method in combination with other types of street-level data such as LiDAR point clouds. This would involve adapting the our model to work with different data formats and modalities. The Kitti dataset (Geiger et al., 2013) would be a suitable, as it contains a lot of different data.

Finally it would be valuable to evaluate the effectiveness of our anonymization method against adversarial attacks or attacks based on deep learning models. We have already mentioned this in Section 2.1.3, this topic has been a recurring theme in our research.

Bibliography

Andreopoulos, Alexander and John Tsotsos (Aug. 2013). "50 Years of object recognition: Directions forward". In: *Computer Vision and Image Understanding* 117, pp. 827–891. DOI: 10.1016/j.cviu.2013.04.005.

Arjovsky, Martin, Soumith Chintala, and Léon Bottou (2017). *Wasserstein GAN*. DOI: 10.48550/ARXIV.1701.07875.

Arnab, Anurag, Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Mans Larsson, Alexander Kirillov, Bogdan Savchynskyy, Carsten Rother, Fredrik Kahl, and Philip Torr (Jan. 2018). "Conditional Random Fields Meet Deep Neural Networks for Semantic Segmentation: Combining Probabilistic Graphical Models with Deep Learning for Structured Prediction". In: *IEEE Signal Processing Magazine* 35, pp. 37–52. DOI: 10.1109/MSP.2017.2762355.

Bay, Herbert, Tinne Tuytelaars, and Luc Van Gool (July 2006). "SURF: Speeded up robust features". In: *Computer Vision-ECCV 2006* 3951, pp. 404–417. DOI: 10.1007/11744023_32.

Bertalmío, Marcelo, Guillermo Sapiro, Vicent Caselles, and C. Ballester (Jan. 2000). "Image inpainting". In: *Proceedings of the ACM SIGGRAPH Conference on Computer Graphics*, pp. 417–424.

Bitouk, Dmitri, Neeraj Kumar, Samreen Dhillon, Peter Belhumeur, and Shree K. Nayar (2008). "Face Swapping: Automatically Replacing Faces in Photographs". In: *ACM Trans. Graph.* 27.3. ISSN: 0730-0301. DOI: 10.1145/1360612.1360638.

Bochkovskiy, Alexey, Chien-Yao Wang, and Hong-Yuan Mark Liao (2020). "YOLOv4: Optimal Speed and Accuracy of Object Detection". In: *CoRR* abs/2004.10934. arXiv: 2004.10934.

Boyle, Michael, Christopher Edwards, and Saul Greenberg (2000). "The Effects of Filtered Video on Awareness and Privacy". In: *Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work*. CSCW '00. Philadelphia, Pennsylvania, USA: Association for Computing Machinery, pp. 1–10. ISBN: 1581132220. DOI: 10.1145/358916.358935.

Brkic, K., I. Sikiric, T. Hrkac, and Z. Kalafatic (July 2017). "I Know That Person: Generative Full Body and Face De-identification of People in Images". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. Los Alamitos, CA, USA: IEEE Computer Society, pp. 1319–1328. DOI: 10.1109/CVPRW.2017.173.

Cai, Zhipeng, Zuobin Xiong, Honghui Xu, Peng Wang, Wei Li, and Yi Pan (2021). "Generative Adversarial Networks: A Survey Towards Private and Secure Applications". In: *CoRR* abs/2106.03785. arXiv: 2106.03785.

Chen, Datong, Yi Chang, and Rong Yan (Dec. 2007). "Tools for Protecting the Privacy of Specific Individuals in Video". In: *EURASIP Journal on Applied Signal Processing* 2007, pp. 107–107. DOI: 10.1155/2007/75427.

Chen, Zhe, Yuchen Duan, Wenhai Wang, Junjun He, Tong Lu, Jifeng Dai, and Yu Qiao (2023). "Vision Transformer Adapter for Dense Predictions". In: *The Eleventh International Conference on Learning Representations*.

Cordts, Marius, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele (2016). "The Cityscapes Dataset for Semantic Urban Scene Understanding". In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Dalal, Navneet and Bill Triggs (June 2005). "Histograms of Oriented Gradients for Human Detection". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2005)* 2.

Demir, Uğur and Gozde Unal (Mar. 2018). *Patch-Based Image Inpainting with Generative Adversarial Networks*.

Devaux, Alexandre, Nicolas Paparoditis, Frederic Precioso, and Bertrand Cannelle (Jan. 2009). "Face blurring for privacy in street-level geoviewers combining face, body and skin detectors". In: *Proceedings of the 11th IAPR Conference on Machine Vision Applications, MVA 2009*, pp. 86–89.

Efros, A.A. and T.K. Leung (1999). "Texture synthesis by non-parametric sampling". In: *Proceedings of the Seventh IEEE International Conference on Computer Vision*. Vol. 2, 1033–1038 vol.2. DOI: 10.1109/ICCV.1999.790383.

European General Data Protection Regulation (2018). European Parliament and Council of the European Union. (accessed: 18.05.2021).

European General Data Protection Regulation (2018). European Parliament and Council of the European Union. (accessed: 23.05.2021).

Felzenszwalb, Pedro, Ross Girshick, David Mcallester, and Deva Ramanan (Sept. 2010). "Object Detection with Discriminatively Trained Part-Based Models". In: *IEEE transactions on pattern analysis and machine intelligence* 32, pp. 1627–45. DOI: 10.1109/TPAMI.2009.167.

Felzenszwalb, Pedro, David McAllester, and Deva Ramanan (2008). "A discriminatively trained, multiscale, deformable part model". In: *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8. DOI: 10.1109/CVPR.2008.4587597.

Flores, Arturo and Serge Belongie (2010). "Removing pedestrians from Google street view images". In: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*, pp. 53–58. DOI: 10.1109/CVPRW.2010.5543255.

Frome, Andrea, German Cheung, Ahmad Abdulkader, Marco Zennaro, Bo Wu, Alessandro Bissacco, Hartwig Adam, Hartmut Neven, and Luc Vincent (Nov. 2009). "Large-scale privacy protection in Google Street View". In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2373–2380. DOI: 10.1109/ICCV.2009.5459413.

Geiger, Andreas, Philip Lenz, Christoph Stiller, and Raquel Urtasun (2013). "Vision meets Robotics: The KITTI Dataset". In: *International Journal of Robotics Research (IJRR)*.

Girshick, Ross, Jeff Donahue, Trevor Darrell, and Jitendra Malik (Nov. 2013). "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation". In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. DOI: 10.1109/CVPR.2014.81.

Girshick, Ross B. (2015). "Fast R-CNN". In: *CoRR* abs/1504.08083. arXiv: 1504.08083.

Goodfellow, Ian (2017). *NIPS 2016 Tutorial: Generative Adversarial Networks*. DOI: 10.48550/ARXIV.1701.00160.

Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Y. Bengio (June 2014). "Generative Adversarial Networks". In: *Advances in Neural Information Processing Systems* 3. DOI: 10.1145/3422622.

Gross, R., L. Sweeney, F. de la Torre, and S. Baker (2006). "Model-Based Face De-Identification". In: *2006 Conference on Computer Vision and Pattern Recognition Workshop*, pp. 161–161. DOI: 10.1109/CVPRW.2006.125.

Gross, Ralph, Latanya Sweeney, Jeffrey Cohn, Fernando De la Torre, and Simon Baker (July 2009). "Face De-identification". In: *Protecting Privacy in Video Surveillance*, pp. 129–146. ISBN: 978-1-84882-300-6. DOI: 10.1007/978-1-84882-301-3_8.

Gulrajani, Ishaan, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville (2017). *Improved Training of Wasserstein GANs*. DOI: 10.48550/ARXIV.1704.00028.

He, Kaiming, Georgia Gkioxari, Piotr Dollár, and Ross Girshick (Mar. 2017). *Mask R-CNN*.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (June 2014). "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37. DOI: 10.1109/TPAMI.2015.2389824.

— (2015). "Deep Residual Learning for Image Recognition". In: *CoRR* abs/1512.03385. arXiv: 1512.03385.

He, Liu, Michael Bleyer, and Margrit Gelautz (2011). *Object Removal by Depth-guided Inpainting*.

Heusel, Martin, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter (2017). "GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium". In: NIPS'17. Long Beach, California, USA: Curran Associates Inc., pp. 6629–6640. ISBN: 9781510860964.

Hrkać, Tomislav, Karla Brkić, and Zoran Kalafatić (2017). *Multi-Class U-Net for Segmentation of Non-biometric Identifiers*.

Hudson, Scott E. and Ian Smith (1996). "Techniques for Addressing Fundamental Privacy and Disruption Tradeoffs in Awareness Support Systems". In: ACM press, pp. 248–257.

Hukkelås, Håkon, Rudolf Mester, and Frank Lindseth (2019). *DeepPrivacy: A Generative Adversarial Network for Face Anonymization*. DOI: 10.48550/ARXIV.1909.04538.

Iizuka, Satoshi, Edgar Simo-Serra, and Hiroshi Ishikawa (July 2017). "Globally and Locally Consistent Image Completion". In: *ACM Trans. Graph.* 36.4. ISSN: 0730-0301. DOI: 10.1145/3072959.3073659.

Isola, Phillip, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros (July 2017). "Image-To-Image Translation With Conditional Adversarial Networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Jain, Anil, Sarat Dass, and Karthik Nandakumar (Aug. 2004). "Can soft biometric traits assist user recognition?" In: *Proceedings of SPIE - The International Society for Optical Engineering* 5404. DOI: 10.1117/12.542890.

Jia, Ning, Victor Sanchez, and Chang-Tsun Li (2017). "Learning Optimised Representations for View-Invariant Gait Recognition". In: *2017 IEEE International Joint Conference on Biometrics (IJCB)*. Denver, CO, USA: IEEE Press, pp. 774–780. DOI: 10.1109/BTAS.2017.8272769.

Johnson, Justin, Alexandre Alahi, and Li Fei-Fei (2016). "Perceptual Losses for Real-Time Style Transfer and Super-Resolution". In: *CoRR* abs/1603.08155. arXiv: 1603.08155.

Karami, Ebrahim, Siva Prasad, and Mohamed Shehata (Nov. 2015). *Image Matching Using SIFT, SURF, BRIEF and ORB: Performance Comparison for Distorted Images*.

Kim, Taehoon and Jihoon Yang (2019). "Latent-Space-Level Image Anonymization With Adversarial Protector Networks". In: *IEEE Access* 7, pp. 84992–84999. DOI: 10.1109/ACCESS.2019.2924479.

Kingma, Diederik and Max Welling (Dec. 2013). "Auto-Encoding Variational Bayes". In: *ICLR*.

Kitahara, Itaru, K. Kogure, and N. Hagita (2004). "Stealth vision for protecting privacy". In: *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.* Vol. 4, 404–407 Vol.4. DOI: 10.1109/ICPR.2004.1333788.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey Hinton (Jan. 2012a). "ImageNet Classification with Deep Convolutional Neural Networks". In: *Neural Information Processing Systems* 25. DOI: 10.1145/3065386.

— (Jan. 2012b). "ImageNet Classification with Deep Convolutional Neural Networks". In: *Neural Information Processing Systems* 25. DOI: 10.1145/3065386.

Li, Jian, Yabiao Wang, Changan Wang, Ying Tai, Jianjun Qian, Jian Yang, Chengjie Wang, Jilin Li, and Feiyue Huang (2018). *DSFD: Dual Shot Face Detector*. DOI: 10.48550/ARXIV.1810.10220.

Li, Tao and Lei Lin (2019). "AnonymousNet: Natural Face De-Identification with Measurable Privacy". In: *CoRR* abs/1904.12620. arXiv: 1904.12620.

Lin, Tsung-Yi, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie (2016). "Feature Pyramid Networks for Object Detection". In: vol. abs/1612.03144. arXiv: 1612.03144.

Lin, Tsung-Yi, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár (2017). "Focal Loss for Dense Object Detection". In: *CoRR* abs/1708.02002. arXiv: 1708.02002.

Liu, Guilin, Fitsum A. Reda, Kevin J. Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro (2018). "Image Inpainting for Irregular Holes Using Partial Convolutions". In: *CoRR* abs/1804.07723. arXiv: 1804.07723.

Liu, Li, Wanli Ouyang, Xiaogang Wang, Paul Fieguth, Jie Chen, Xinwang Liu, and Matti Pietikäinen (Sept. 2018). *Deep Learning for Generic Object Detection*.

Liu, Weiyang, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song (2017). *SphereFace: Deep Hypersphere Embedding for Face Recognition*. DOI: 10.48550/ARXIV.1704.08063.

Liu, Ze, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo (Oct. 2021). "Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10012–10022.

Lowe, David (Jan. 2001). "Object Recognition from Local Scale-Invariant Features". In: *Proceedings of the IEEE International Conference on Computer Vision* 2.

Lugmayr, Andreas, Martin Danelljan, Andrés Romero, Fisher Yu, Radu Timofte, and Luc Van Gool (2022). "RePaint: Inpainting using Denoising Diffusion Probabilistic Models". In: *CoRR* abs/2201.09865. arXiv: 2201.09865.

Luo, Juan and Gwun Oubong (Oct. 2009). "A comparison of sift, pca-sift and surf". In: *International Journal of Image Processing* 3.

Maximov, Maxim, Ismail Elezi, and Laura Leal-Taixe (June 2020). "CIAGAN: Conditional Identity Anonymization Generative Adversarial Networks". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. DOI: 10.1109/cvpr42600.2020.00549.

McPherson, Richard, Reza Shokri, and Vitaly Shmatikov (2016). "Defeating Image Obfuscation with Deep Learning". In: *CoRR* abs/1609.00408. arXiv: 1609.00408.

Mirza, Mehdi and Simon Osindero (2014). *Conditional Generative Adversarial Nets*. DOI: 10.48550/ARXIV.1411.1784.

Mistry, Dr and Asim Banerjee (Mar. 2017). "Comparison of Feature Detection and Matching Approaches: SIFT and SURF". In: *GRD Journals- Global Research and Development Journal for Engineering* 2, pp. 7–13.

Miyato, Takeru, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida (2018). "Spectral Normalization for Generative Adversarial Networks". In: *CoRR* abs/1802.05957. arXiv: 1802.05957.

Najibi, Mahyar, Pouya Samangouei, Rama Chellappa, and Larry Davis (2017). *SSH: Single Stage Headless Face Detector*. DOI: 10.48550/ARXIV.1708.03979.

Nazeri, Kamyar, Eric Ng, Tony Joseph, Faisal Z. Qureshi, and Mehran Ebrahimi (2019). "EdgeConnect: Generative Image Inpainting with Adversarial Edge Learning". In: *CoRR* abs/1901.00212. arXiv: 1901.00212.

Newton, E.M., L. Sweeney, and B. Malin (2005). "Preserving privacy by de-identifying face images". In: *IEEE Transactions on Knowledge and Data Engineering* 17.2, pp. 232–243. DOI: 10.1109/TKDE.2005.32.

Ouyang, Xi, Yu Cheng, Yifan Jiang, Chun-Liang Li, and Pan Zhou (2018). *Pedestrian-Synthesis-GAN: Generating Pedestrian Data in Real Scene and Beyond*. DOI: 10.48550/ARXIV.1804.02047.

Padilla-López, José Ramón, Alexandros Andre Chaaraoui, and Francisco Flórez-Revuelta (2015). "Visual privacy protection methods". In: *Expert Systems with Applications* 42.9, pp. 4177–4195. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2015.01.041.

Papageorgiou, C.P., Michael Oren, and Tomaso Poggio (Feb. 1998). "General framework for object detection". In: vol. 6: pp. 555–562. ISBN: 81-7319-221-9. DOI: 10.1109/ICCV.1998.710772.

Pathak, Deepak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros (2016). "Context Encoders: Feature Learning by Inpainting". In: DOI: 10.48550/ARXIV.1604.07379.

Qiao, Siyuan, Liang-Chieh Chen, and Alan Yuille (June 2021). "DetectoRS: Detecting Objects with Recursive Feature Pyramid and Switchable Atrous Convolution". In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, pp. 10208–10219. DOI: 10.1109/CVPR46437.2021.01008.

Radford, Alec, Luke Metz, and Soumith Chintala (2015). *Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks*. DOI: 10.48550/ARXIV.1511.06434.

Redmon, Joseph, Santosh Divvala, Ross Girshick, and Ali Farhadi (June 2015). *You Only Look Once: Unified, Real-Time Object Detection*.

Redmon, Joseph and Ali Farhadi (2017). "YOLO9000: Better, Faster, Stronger". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6517–6525. DOI: 10.1109/CVPR.2017.690.

— (Apr. 2018). *YOLOv3: An Incremental Improvement*.

Reid, D.A., S. Samangooei, C. Chen, M.S. Nixon, and A. Ross (2013). "Chapter 13 - Soft Biometrics for Surveillance: An Overview". In: *Handbook of Statistics*. Ed. by C.R. Rao and Venu Govindaraju. Vol. 31. Handbook of Statistics. Elsevier, pp. 327–352. DOI: https://doi.org/10.1016/B978-0-444-53859-8.00013-8.

Ren, Shaoqing, Kaiming He, Ross Girshick, and Jian Sun (June 2015). "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39. DOI: 10.1109/TPAMI.2016.2577031.

Ren, Zhongzheng, Yong Jae Lee, and Michael S. Ryoo (Sept. 2018). "Learning to Anonymize Faces for Privacy Preserving Action Detection". In: *Proceedings of the European Conference on Computer Vision (ECCV)*.

Ribaric, S., Aladdin Ariyaeeinia, and Nikola Pavesic (June 2016). "De-identification for privacy protection in multimedia content". In: *Signal Processing: Image Communication* 47. DOI: 10.1016/j.image.2016.05.020.

Ronneberger, Olaf, Philipp Fischer, and Thomas Brox (2015). "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *CoRR* abs/1505.04597. arXiv: 1505.04597.

Sadimon, Suriati Bte, Mohd Shahrizal Sunar, Dzulkifli Mohamad, and Habibollah Haron (2010). "Computer Generated Caricature". In: *2010 International Conference on Cyberworlds*, pp. 383–390. DOI: 10.1109/CW.2010.33.

Saharia, Chitwan, William Chan, Huiwen Chang, Chris A. Lee, Jonathan Ho, Tim Salimans, David J. Fleet, and Mohammad Norouzi (2022). *Palette: Image-to-Image Diffusion Models*.

Sudre, Carole H., Wenqi Li, Tom Vercauteren, Sébastien Ourselin, and M. Jorge Cardoso (2017). "Generalised Dice overlap as a deep learning loss function for highly unbalanced segmentations". In: *CoRR* abs/1707.03237. arXiv: 1707.03237.

Szegedy, Christian, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna (2015). *Rethinking the Inception Architecture for Computer Vision*. DOI: 10.48550/ARXIV.1512.00567.

Tan, Mingxing, Ruoming Pang, and Quoc Le (2019). "EfficientDet: Scalable and Efficient Object Detection". In: vol. abs/1911.09070. arXiv: 1911.09070.

Tang, Lei (1997). "Methods for Encrypting and Decrypting MPEG Video Data Efficiently". In: *Proceedings of the Fourth ACM International Conference on Multimedia*. MULTIMEDIA '96. Boston, Massachusetts, USA: Association for Computing Machinery, pp. 219–229. ISBN: 0897918711. DOI: 10.1145/244130.244209.

Tansuriyavong, Suriyon and Shin-ichi Hanaki (2001). "Privacy Protection by Concealing Persons in Circumstantial Video Image". In: *Proceedings of the 2001 Workshop on Perceptive User Interfaces*. PUI '01. Orlando, Florida, USA: Association for Computing Machinery, pp. 1–4. ISBN: 9781450374736. DOI: 10.1145/971478.971519.

Uijlings, Jasper, K. Sande, T. Gevers, and A.W.M. Smeulders (Sept. 2013). "Selective Search for Object Recognition". In: *International Journal of Computer Vision* 104, pp. 154–171. DOI: 10.1007/s11263-013-0620-5.

Uittenbogaard, Ries, Clint Sebastian, Julien Vijverberg, Bas Boom, Dariu Gavrila, and Peter With (2019). "Privacy Protection in Street-View Panoramas using Depth and Multi-View Imagery". In: vol. abs/1903.11532. arXiv: 1903.11532.

van Zoonen, Liesbet (2016). "Privacy concerns in smart cities". In: *Government Information Quarterly* 33.3. Open and Smart Governments: Strategies, Tools, and Experiences, pp. 472–480. ISSN: 0740-624X. DOI: https://doi.org/10.1016/j.giq.2016.06.004.

Viola, Paul and Michael Jones (2001). "Rapid object detection using a boosted cascade of simple features". In: *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*. Vol. 1, pp. I–I. DOI: 10.1109/CVPR.2001.990517.

Wang, Tengfei, Hao Ouyang, and Qifeng Chen (2021). "Image Inpainting with External-internal Learning and Monochromic Bottleneck". In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5116–5125.

Wang, Zhou, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli (2004). "Image quality assessment: from error visibility to structural similarity". In: *IEEE Transactions on Image Processing* 13.4, pp. 600–612. DOI: 10.1109/TIP.2003.819861.

Woo, Sanghyun, Jongchan Park, Joon-Young Lee, and In So Kweon (2018). "CBAM: Convolutional Block Attention Module". In: *CoRR* abs/1807.06521. arXiv: 1807.06521.

Xiao, Jing, Liang Liao, Qiegen Liu, and Ruimin Hu (July 2019). "CISI-net: Explicit Latent Content Inference and Imitated Style Rendering for Image Inpainting". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 33.01, pp. 354–362. DOI: 10.1609/aaai.v33i01.3301354.

Yang, Chao, Xin Lu, Zhe Lin, Eli Shechtman, Oliver Wang, and Hao Li (2016). "High-Resolution Image Inpainting using Multi-Scale Neural Patch Synthesis". In: *CoRR* abs/1611.09969. arXiv: 1611.09969.

Yang, M., N. Bourbakis, and Shujun Li (2004). "Data-image-video encryption". In: *IEEE Potentials* 23.3, pp. 28–34. DOI: 10.1109/MP.2004.1341784.

Yang, Wei, Ping Luo, and Liang Lin (2014). "Clothing Co-parsing by Joint Image Segmentation and Labeling". In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3182–3189. DOI: 10.1109/CVPR.2014.407.

Ye, Mang, Jianbing Shen, Gaojie Lin, Tao Xiang, Ling Shao, and Steven C. H. Hoi (2020). *Deep Learning for Person Re-identification: A Survey and Outlook*. DOI: 10.48550/ARXIV.2001.04193.

Youzi, Xiao, Zhiqiang Tian, Jiachen Yu, Yinshu Zhang, Shuai Liu, Shaoyi Du, and Xuguang Lan (Sept. 2020). "A review of object detection based on deep learning". In: *Multimedia Tools and Applications* 79. DOI: 10.1007/s11042-020-08976-6.

Yu, Fisher and Vladlen Koltun (2015). *Multi-Scale Context Aggregation by Dilated Convolutions*. DOI: 10.48550/ARXIV.1511.07122.

Yu, Jiahui, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas Huang (2019). "Free-Form Image Inpainting With Gated Convolution". In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4470–4479. DOI: 10.1109/ICCV.2019.00457.

Yu, Jiahui, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S. Huang (2018). *Generative Image Inpainting with Contextual Attention*. DOI: 10.48550/ARXIV.1801.07892.

Zaidi, Syed Suleman Abbas, Mohammad Samar Ansari, Asra Aslam, Nadia Kanwal, Mamoona Naveed Asghar, and Brian Lee (2022). "A Survey of Modern Deep Learning based Object Detection Models". In: *Digit. Signal Process.* 126, p. 103514.

Zhang, Cha, Yong Rui, and Li-wei He (Nov. 2006). "Light Weight Background Blurring for Video Conferencing Applications". In: *Proceedings / ICIP ... International Conference on Image Processing*, pp. 481–484. DOI: 10.1109/ICIP.2006.312498.

Zhang, Kaipeng, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao (Oct. 2016). "Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks". In: *IEEE Signal Processing Letters* 23.10, pp. 1499–1503. DOI: 10.1109/lsp.2016.2603342.

Zou, Zhengxia, Zhenwei Shi, and Yuhong Guo (May 2019). *Object Detection in 20 Years*.

Appendices

Appendix A

Confusion matrix

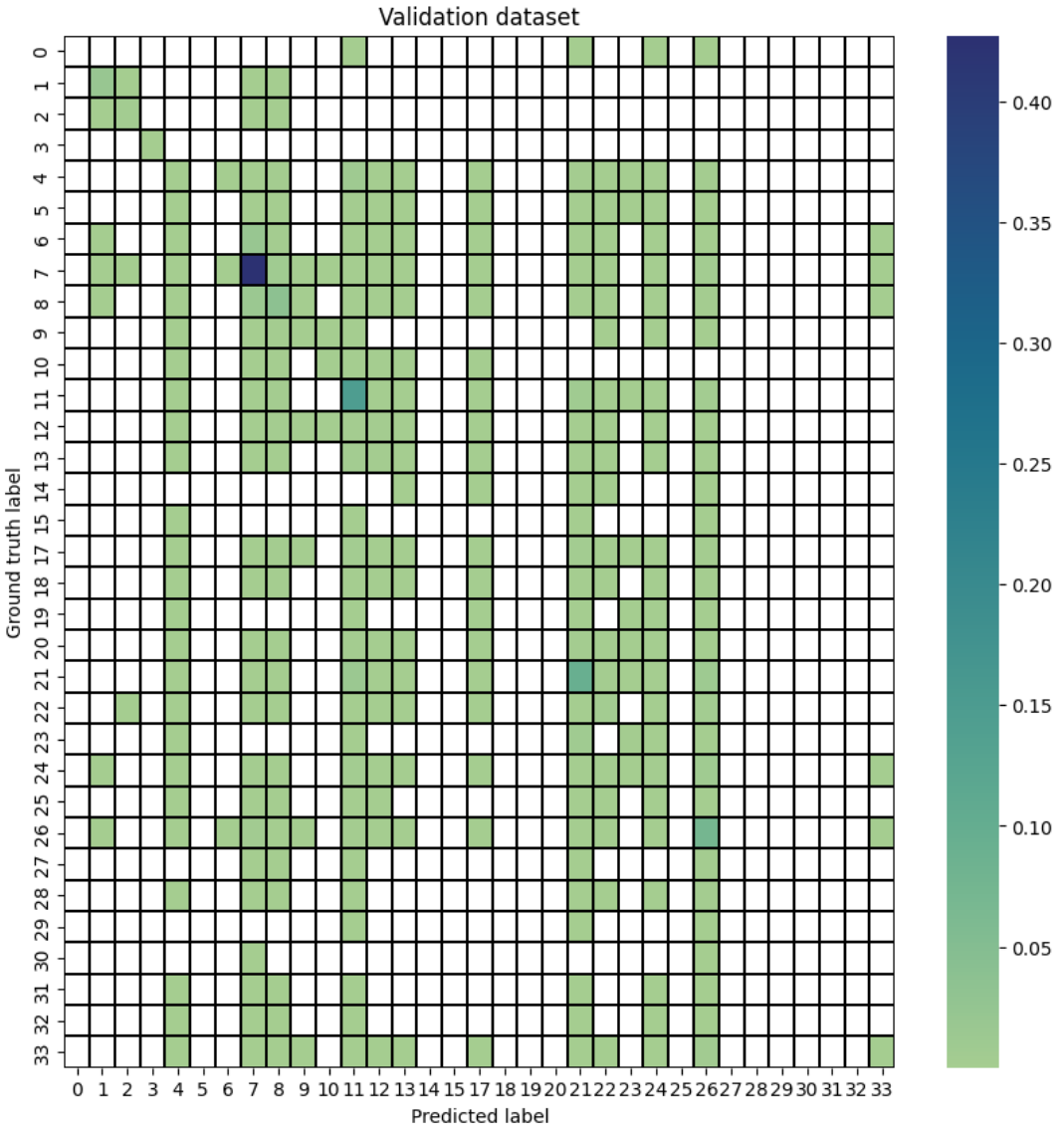


Figure A.1: Confusion matrix of the semantic inpainting model calculated for the validation data. The y-axis represents the ground truth labels, while the x-axis represents the predicted labels.