DeepClean: A Robust Deep Learning Approach for Autonomous Vehicle Camera Data Privacy

Olayinka Adeboye

Submitted in Partial Fulfilment of the Requirements of the Degree of
Doctor of Philosophy

Doctor of Philosophy
in
Cyber Security

University of Salford
March 2023
Manchester, UK

ABSTRACT

Autonomous Vehicles (AVs) generate several forms of tracking data, such as geolocation, distance, and camera data. The utility of these data, especially camera data for computer vision projects, has contributed to the advancement of high-performance self-driving applications. However, location inference attacks, which involve extracting knowledge from camera data to track and estimate user locations are potential privacy threats to AV-generated camera data. Recently, a few studies investigated privacy-preserving approaches for AV-generated camera data using powerful generative models such as Variational Auto Encoder (VAE) and Generative Adversarial Network (GAN). However, the related work considered a weak geo-localisation attack model, which leads to weak privacy protection against stronger attack models.

This study develops LIFT (Location InFerence aTtack), a robust geo-localisation technique to exploit subjects' location privacy in a GAN-based camera dataset. LIFT's performance is evaluated on a 200k Google Street view as a reference dataset and 500 distorted image datasets as test query data. The result obtained show that the localisation accuracy of LIFT outperforms the benchmark techniques by 20%.

To efficiently address AV camera data privacy preservation, DeepClean is proposed in this thesis. DeepClean combines VAE and private clustering to learn distinct labelled object structures of the image data in clusters. It then generates a more visual representation of the non-private object clusters, e.g., roads, and distorts the private object areas using a private Gaussian Mixture Model (GMM) to learn distinct cluster

structures of the labelled clusters. The synthetic images generated from DeepClean guarantee privacy and resist robust location inference attacks (such as LIFT) by less than 4% localisation accuracy. The image utility level of the non-private object areas is comparable to the benchmark studies.

Keywords: Autonomous vehicle, Data privacy, Data utility, Deep clustering, Generative model, Differential privacy.

The true sign of intelligence is not knowledge but imagination..... Albert Einstein

ACKNOWLEDGMENT

I am grateful to my supervisor, Dr Tooska Dargahi, for her expert guidance and support throughout my PhD research. I also acknowledge the constant training and encouragement from Prof. Mo Saraee and Dr Meisam Babaie. Without their guidance and corrections, I would never have been able to complete the research.

I am grateful to my lovely wife and kids for their family love and inspiration that kept me going. All of them, including my parents, contributed to financing and counselling.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

1.1 Introduction

Autonomous vehicles (AV) onboard sensory devices generate diverse datasets [1]. These datasets include camera data (images and videos of street views showing road objects in a city), distance data from Lidar and Radar sensors, and Global Positioning Systems (GPS) trajectory data. The captured datasets are required for several functional and non-functional processes [2, 3]. For instance, the captured visual images and videos can be used for accident claims and training auto-driving deep learning models (e.g., for object detection and recognition [4–6]). Also, real-time data analysis on in-vehicle data is used for performance evaluation purposes [7, 8]. This rich dataset could be held inside the vehicle or sent to external storage, such as cloud [2].

One of the main concerns regarding AV-generated data is users' privacy [9]. Camera data contain several visual and context-rich features that can be extracted and geo-localised. Several studies have shown how over-needed location information in images, such as background buildings, landmarks, road signs and markings, and surrounding vegetation improve image matching and geo-localisation [10]. Suppose we assume an attacker can get unauthorised access to the stored camera data in the internal or external storage. In that case, she (referring to the attacker throughout the thesis) can perform a

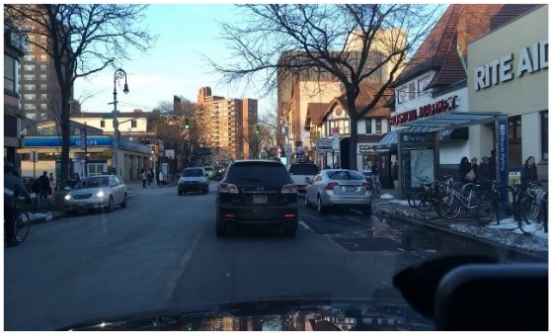
location inference attack using geo-localisation techniques. This attacker may be able to infer sensitive information, such as the user's home/work address and past/future travel patterns, which leads to a location privacy breach.

Figure 1.1 shows an example of a location inference attack. If an attacker has access to both Figure 1.1 (a) and (b), she can learn that the target's vehicle has passed through Queen's Tower (Figure 1.1 (a)). By getting access to more images and videos from the target's vehicle with timestamp correlations (Figure 1.1 (b)) she can perform geo-localisation and predict the trajectories and journey patterns, refer to Figure 1.1 (c) for an example.

Researchers have proposed several ways of distorting private objects in a dataset to mitigate location inference attacks on AV-generated camera data. Traditional image obfuscation methods such as pixelating, masking, blurring, and silhouette produce blurry outputs because of their application effects on individual pixels in the image. Their output cannot sufficiently satisfy the privacy-preserving goals for AV image storage and processing. Recently, *Xiong et al.* proposed ADGAN, in which they use both Variational Autoencoder (VAE) and Generative Adversarial Network (GAN) to generate privacy-preserving camera datasets [14]. They have considered a weak attack model under Multi-KNN (i.e., multiple k nearest neighbour) feature matching geo-localisation approach. Multi-KNN [13] was also used in other research studies for geo-localisation, such as [12, 15]. Secondly, their method (ADGAN) does not provide a formal privacy guarantee to prove the resistance to a robust attack.

[a]

[b]



[c]

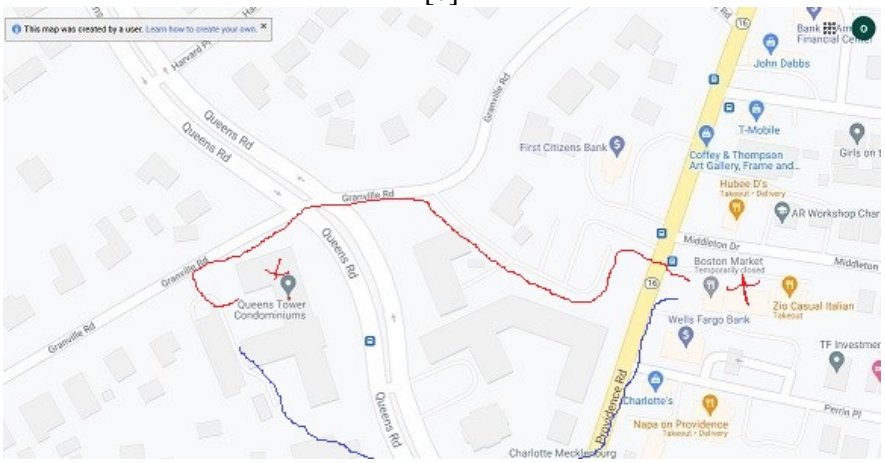


Figure 1.1: An example to show a geo-localisation attack. (a) Camera data of the target's vehicle near Queen's Tower, (b) Camera data of the target's vehicle near Boston Market. (c) Leaked trajectory information between both locations. All the images are extracted from Berkeley AV Open-source data [3]

Schindler et al. organised image features as a bag of words and arranged them in a vocabulary tree for image matching [12]. Their approach is inefficient for processing large image features and computationally too slow. *Zamir et al.* improved the computational efficiency of feature matching using a generalised minimum clique problem [13]. However, their formulation of a fixed nearest neighbour selection algorithm limits the number of matching features and hence does not allow for image matching improvement. *Zemene et al.* [11] designed a more robust geo-localisation system to localise street view images with higher performance compared to the previous studies, and other image geo-localisation approaches [16, 17]. The improved feature matching approach in [11] is based on returning a dynamic nearest neighbour of the reference images using dominant set clustering, which outperforms the approaches based on multi-KNN with a fixed value for k. However, improved geo-location estimate increases the image matching performance with the cost of increased potential privacy threats.

The notion of robust image matching is the main motivation in this thesis to improve the geo-localisation method in [11] and propose a strong attack model against AV-generated camera data. Distinctive nearest neighbours (NNs) are selected from each image and organised in a database to improve feature matching. A pairwise clustering method matches similar image features by an optimised quadratic and similarity function, as explained in Chapter 4. Considering this localisation attack, the following research questions are drawn:

1) What features in an image could be manipulated to decrease the similarity between

an original image and its distorted version to improve privacy?

2) Can a privacy-preserving technique for AV camera data sufficiently balance the privacy-utility trade-off to suit several data use cases?

To provide answers to these research questions, DeepClean is proposed and developed.

(a) DeepClean is a deep clustering approach which combines VAE with GMM clustering methods to improve the privacy-utility trade-off. It proposes a solution for learning and controlling the visual representation of objects in an image. Two labels, i.e., private and non-private labels, are considered for classifying the image's objects. Private objects are those that could significantly help in the geo-localisation process, such as buildings, pedestrians, vehicles, and road signs. Deep clustering is used to learn, separate and then distort those clusters that include private objects while retaining the underlying structure of the non-private object areas (e.g., roads). The GMM clustering method is used for learning clusters of objects in high-dimensional image data that are well-separated to enforce the privacy/utility requirements. The private GMM method satisfies differential privacy, which has been the de facto standard for protecting data privacy in the statistical analysis of sensitive data.

(b) DeepClean uses the VAE data generation technique to produce high-dimensional image samples without directly operating on original data. The VAE approach is flexible for 3D street view models and traffic analysis applications. ADGAN II [14] adopts the VAE to improve the data generation performance from dis-

tributional assumptions, in contrast to image-dependent processing applications such as UNIT in ADGAN I [18].

(c) DeepClean utilises an encoder and decoder model. Its encoder model encodes data by partitioning it into object clusters using the proposed private GMM algorithm. A function of the algorithm learns a supervised clustering task and accurately partitions the clusters into private and non-private object parts by using mask binary code as a key. The learned private object clusters are distorted by injecting Gaussian noise into their cluster centres. This approach ensures that we can efficiently preserve privacy in the private cluster areas without affecting too much visual quality of the non-private object areas.

(d) DeepClean's decoder model decodes the resulting high-dimensional feature representation from the encoder network into observable samples using a deep neural network. The model optimisation is achieved by maximising the expected lower bound of the VAE system.

1.2 Motivation

With the advancements in sensor technology and automotive systems, the idea of a functional AV constantly operating on the roads is becoming more realistic. While their operations entirely depend on data collection, storage, processing, smart communication, data sharing, and external storage, information security and privacy risks pose major concerns to the vehicle's data. A study identified that the AV platform and applications would attract cyber-attacks from several malicious actors [19]. *Petit et al.*

claimed that security attacks on AV systems would be more of sabotage (compromising a system's operation) than an act of espionage (gaining unauthorised system access for information). Attack types may include the disruption of AV communication channels and sensors, virus attacks and the system's programs to cause a denial of service and traffic disruptions. Data privacy concerns on the AV mainly emphasised the misuse of the data by parties who have access to it, giving rise to several threats such as stalking, surveillance, and inference of subjects associated with the AV [20]. A good example of an AV privacy breach was depicted in a 2022 movie titled "The Takeover", where state-sponsored threat actors exploited the bug in an AV's facial recognition application to collect users' facial biometrics for criminal activities through Deepfakes.

After researching these potential security and privacy concerns on AVs with recent cyber hacks and constant attacks on information systems in mind, there is a need to analyse gaps and propose AV-related controls critically. These controls must ensure adequate privacy protections against several threats. Since this research focuses on AV camera data privacy, the motivation lies in analysing location privacy threats and developing a privacy-preserving technique for AV camera data processing. Many traditional camera data processing methods have been used in several works (to be discussed in Chapter 3) to protect privacy. However, their statistical applications affect the entire keypoints in the image, resulting in either less privacy protection with a lower parameter or too blurry output with a higher parameter. This challenge presents a problem where the privacy/utility goal is to produce an indistinguishable image from an original one. The attempted solution to this challenge works by predicting pixels from

pixels and translating the representation of an image into another from a model trained with enough datasets where metrics measure dissimilarities in private objects. Utility metrics measure similarities in non-private objects in the images.

Variational autoencoder (VAE) and Generative Adversarial Networks (GAN) have been used to achieve this goal; however, this study investigates how a robust attack process exploits images generated by these methods and proposes a stronger generative method. Therefore, this thesis is motivated to design a private generative technique that combines VAE with a privacy-amenable clustering method, Gaussian Mixture Model (GMM). This approach will ensure stricter privacy protection in private object areas using differential privacy for image indistinguishability and good utility preservation in non-private object areas by the model. The images generated from the resulting model will be more strongly immune to location inference attacks than what was achieved in previous works and, at the same time, preserve good overall utility.

1.3 Aim and Objectives

This work aims to protect AV camera data against location inference attacks. The generated data from the proposed model will enhance privacy-preserving data storage and processing. Hence, the research objectives are as follows:

- Conduct a literature review on AV camera data and other external data sources to analyse privacy concerns and efficient privacy-preserving techniques with generative models to address camera data privacy/utility challenges.

- Build threat models to conceptualise the risk assumptions and build a threat tree to detail all the privacy threats with the corresponding system's preconditional vulnerabilities.

- Collect the research dataset (i.e., street view images and videos) from open-source repositories such as Cityscape [165], Berkeley [91], and Lynx [91].

- Define a robust location inference attack model and perform an attack on the collected data to demonstrate the effect and significance of the attack.

- Perform image-to-image modelling to represent the real distribution of the original data and produce high-dimensional realistic images that are indistinguishable from the original data. To achieve this, a privacy-preserving generative technique called DeepClean will be proposed.

- Compare and evaluate the model results with other benchmark models.

- Validate results to demonstrate that they meet the goal: a balanced camera data privacy/utility trade-off.

1.4 Contributions

The main contributions of this thesis are as follows:

1. A robust location inference attack technique, LIFT, is presented and developed. The technique localises distorted query images (e.g., generated using Auto-driving GAN) from a reference database of large-scale geo-tagged images. It formalises

a novel distinctive nearest neighbour selection for stable feature detection and extraction. Image matching and localisation are improved using a pairwise clustering approach to handle large graph cliques. The developed attacking tool (LIFT) will be used in a later chapter to evaluate the privacy performance of synthetic camera data samples from the benchmark generative models.

2. DeepClean, a privacy-preserving generative technique for AV camera data, is developed. It combines a private Gaussian Mixture Model with a Variational Autoencoder to learn high-dimensional feature representations of images as a supervised private and non-private cluster task. Then train the cluster outputs on a Variational Auto Encoder to generate more privacy-protected samples from the developed model.

3. Through real-world publicly available data experiments, DeepClean learns more features in an image, variably controls privacy/utility requirements and generates more privacy-preserved image data considering the location privacy threat.

1.5 Thesis Outline

The structure of this thesis is presented as follows, with each chapter outlining the topic and a brief explanation.

Chapter 2. Background: This chapter introduces the autonomous vehicle, its data collection, storage, and processing, and gives a basic definition of privacy terminologies and an overview of several generative and privacy techniques used in this thesis.

Chapter 3. Literature Review: This chapter focuses on an in-depth analysis of research works performed in AV privacy, efficient privacy-preserving methods, and generative models. It develops arguments based on the problems and the choice of privacy solution for this challenge. The related works investigated the applications of privacy-preserving mechanisms for AVs and connected vehicles and their sensor data for protection against location inference attacks. Hence, it concludes with the suitable methods for AV data privacy protection.

Chapter 4. Location Inference Attacks on AV Camera Data – Threat Model: This chapter presents LIFT, the location inference approach to street view images, as a threat model. It explores geo-localisation techniques for recognising features in a location scene and proposes an approach to localise distorted image data. This chapter also reviews the data privacy protection framework that addresses AV privacy threats. This framework will guide the analysis steps in the remaining chapters.

Chapter 5. Formal Analysis of Generative Clustering Models and DeepClean Model: This chapter outlines the formal analysis and notations used in formulating the proposed technique. This chapter introduces the proposed technique, DeepClean. It explains the various components and the designed algorithm.

Chapter 6. Experimental Results and Findings: This chapter evaluates DeepClean using various metrics and compares DeepClean with benchmark models. It validates the privacy/utility performance of the approach.

Chapter 7. Conclusion and Future Work: This chapter summarises the main findings

of the research and the challenges faced. It also discusses the possible future direction to tackle the challenge.

Chapter 2

Background

2.1 Introduction

An autonomous vehicle is a driverless car that can make intelligent driving decisions without or with little human intervention [21]. It is referred to as a connected autonomous vehicle when it exchanges data with other road-side devices, e.g., road-side unit (RSU), Vehicle to Vehicle (V2V), Vehicle to Everything (V2X) [22]. The difference between autonomous and automated is the degree to which the system depends on human input [23]. The Society of Automotive Engineers (SAE) categorically listed five AV autonomy levels, as in Figure 2.1 [24].

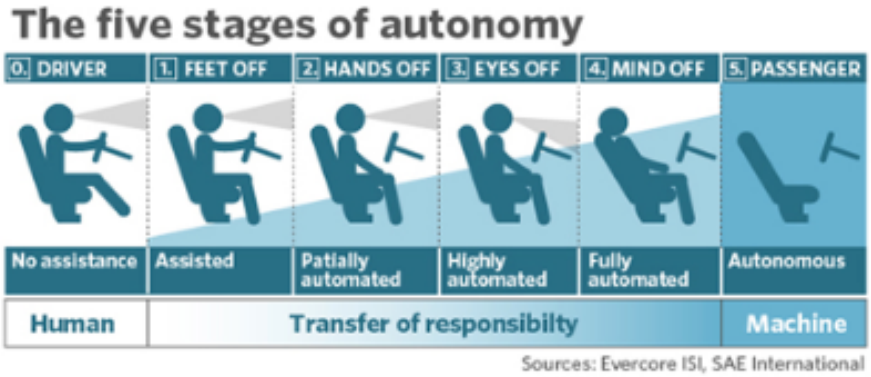


Figure 2.1: Five levels of vehicle autonomy [24]

By trends, car driving has evolved from manual operations involving hectic human driving chores to automatic or automated driving, which introduced self-driving fea-

tures like auto-braking, parking assistance, etc. Now, vehicles are more equipped with modern sensors to collect physical and environmental data, e.g., images and videos, GPS location, Lidar and Radar data, traffic control messages, etc. These data fundamentally serve several purposes, for example, as an input to the control systems where functions to operate the vehicle are initiated. Another purpose is to enable and enhance data analytics to develop more efficient systems and services.

Societal benefits of AVs over traditional driving include more safety, reduced traffic accidents caused by human error, relieved driving chores, lower fuel consumption, and less toxic exhaust [25]. According to a UK forecast, the market of AVs is estimated to be worth $28bn in 2035, capturing 3% of the $907bn global markets. However, it is estimated that UK jobs in the manufacture and assembly of AVs would reach 27,400 in 2035 compared to around 151,000 people currently employed in car manufacturing [26]. Regarding data generation, the AV will generate large volumes of data each day, estimated at 40 terabytes, which approximates 2600GB more than a person's internet use in a day [31].

In the recent data market, data is tipped to be the next oil [26], and AV data will be a major contributor. Already there are higher demands for access to the in-vehicle data (which are data processed inside the vehicle) in specific areas such as insurance liability [27], legal use by the government [28], service provision, road maintenance [29], academic research, etc. Access to in-vehicle data will undoubtedly play an essential role in developing the technology, encouraging innovative in-vehicle service applications, and fostering academic research. Some manufacturers already implement vari-

ous in-vehicle data access solutions to make data available for service providers. For example, Google Waymo uses the extended data server to provide access to its vehicle data through its website [30], Navya uses the Onboard Application Platform (OBAP) through NAVYALEAD (software used by NAVYA) to access real-time data and the Onboard Diagnostic (OBD-II) port to access diagnostic data [32]. It is also reported that some vehicle manufacturers collaborate with cloud service providers such as Microsoft Azure and IBM Blue Mix to manage a neutral server for a shared data server and business-to-business (B2B) marketplace [33]. Therefore, AV data are remotely uploaded to the data servers and will be accessible through web portals for more flexible use, like data analytics.

Despite the lucrative innovation, the potential societal benefits, and the huge data market of the AV, concerns have been expressed mainly in safety operations, cybersecurity, and data privacy [34]. Much AV research and development has focused on optimal safety operations [35]. Which mainly covers system developments for safety and convenience, mitigating flaws in designs, functions, and sensors to reduce road accidents [23, 25, 36–41]. Cybersecurity framework development also guides the safety operations of the AV and addresses the mitigation of significant security risks [42, 43]. The framework proposes new authentication protocols for the AV communication layers that interconnect the Electric Control Unit (ECUs), Vehicle Ad hoc Network (VANET), and Dedicated Short-range Communication [44–47]. Many other works on VANET authentication and anonymisation explore traditional cryptographic encryption and pseudonyms techniques for in-vehicle communications and Vehicle to

Vehicle (V2V) communication beacons. Other works on safe AV data communication and authentication focus on lightweight implementations to reduce complexity and privacy issues in V2V, such as linking changing pseudonyms in changing zones. Many methods and frameworks on AV cybersecurity proposed safer data communication and secure connectivity across all Vehicle to Everything (V2X) applications.

Since this thesis focuses on data privacy, it addresses efforts to protect in-vehicle and out-vehicle data (i.e., data transmitted out of the vehicle for further processing). The steps to protect the data implement several technical controls to privacy risks and enhance compliance with privacy regulations, e.g., General Data Protection Regulation (GDPR) [48, 49], which enforces the lawful processing of AV data. Organisations are fined for non-compliance for violating the duty to implement and maintain reasonable security and privacy procedures.

Table 2.1 briefly describes the privacy terminologies used throughout this thesis. Then several privacy threats to private data are explained in Table 2.3. Privacy risk analysis identifies the most significant threats and risks and suggests efficient data protection controls.

Before collecting, storing, processing, and publishing any private data containing sensitive information, a disclosure risk analysis is initially performed to identify the potential privacy risks to the data. Afterwards, an appropriate privacy-preserving technique, e.g., anonymisation or perturbation, is chosen and applied to maintain a balance between minimising risk and maintaining access to accurate datasets [51]. Meanwhile,

observations relating to the dataset's properties and system efficiencies affect these techniques' performances and the application choice. The properties may include the volume and dimensionality of the dataset, highly sensitive attributes, attribute relationships and correlation, data use-cases, and system performance. Hence, sensitive vehicle data requires reasonable privacy protection, usually enforced by applying several privacy-preserving techniques such as anonymisation, synthetization, and encryption.

Anonymisation techniques based on generalisation and suppression, e.g., k-anonymity and its variations, are simple to apply and perform better on a smaller dataset with fewer attributes [52]. However, attribute privacy is not convincingly guaranteed given the unrestrictive background knowledge of an adversary. Privacy loss may increase given datasets with large volumes and dimensionality.

A state-of-the-art privacy-preserving approach based on Differential privacy (DP) works by producing a privacy-guaranteed output dataset. With an intuition that the output obtained is irrespective of whether a specific subject's record is present or not in the input dataset, which differs by one input [53]. DP addresses some privacy weaknesses that are vulnerable to some attacks in the generalisation-based methods, e.g., linkage, correlation, and reconstruction attacks. Other aspects of DP include its strong privacy notion without limiting the adversary's background knowledge and, more interestingly, its formal mathematical quantification of privacy and utility loss. However, under a strict privacy budget, the approach may significantly affect the data's usefulness, thus, undermining the essence of data analysis [54]. The data properties, such as the format, correlation, dependence, and dimensionality, may affect the performance of a DP tech-

nique. Thus, the properties mentioned above are criteria to consider when analysing the performance of this technique. Prior works proposed the encryption of datasets for analysis to improve data accuracy under minimum privacy considerations [55]. The approach has challenges, including computational complexity as the dataset grows, key maintenance, and result inference risks. This study leans towards sythentization because of its wide application use, efficiency, and scalability for image data analytics and privacy. Statistical agencies also adopt them to produce synthetic datasets because of their statistical usefulness and confidential information disclosure prevention [56].

2.2 AV Components and Functionality

With the huge volume of data, it is informative to know the types of data the sensors generate. Therefore, this section discusses various in-vehicle components that take part in data collection, storage, and processing for the autonomous functions of the system. The Section provides a brief description and properties of the components. This information will help to build a data flow diagram in Chapter 4 and identify the potential privacy threats to the data.

Sensors

AVs use many sensors to collect specific information from their surroundings. Each sensor has its operational capabilities and collects specific physical data. AV manufacturers may use different combinations of sensors. Table 2.5 lists different sensors with their functions and the data type they collect.

- Light Detection and Ranging (LiDAR): This short-range optical device can measure the precise distance to its target and use its laser light to detect and track objects. Its distinct properties from other sensors are producing 3D maps of the environment and detecting stationary objects. It is situated on the vehicle's rooftop but sometimes installed on the side.

- Radio Detection and Ranging (RADAR): transmits radio waves to detect short-range or long-range objects and measures moving objects' distances and relative speed. Despite its unreliable detection of uniform objects, they have a robust functionality advantage over other sensors. Distinct properties include its undisrupted performance in harsh weather conditions, long-range detection, and detecting transparent objects. Depending on the manufacturer's specs, the radar sensor is usually located at the front and rear of the vehicle.

- Cameras: Regardless of the type of camera, they all recognise uniform and moving objects within their visibility. Meanwhile, the major AV types are the complementary metal-oxide semiconductor (CMOS), charge-coupled device (CCD), and Monochrome and stereo cameras for colour identification of traffic signs and traffic lights. Out of the sensors the AV uses, the camera sensor is the only sensor capable of recognising and distinguishing objects based on a machine-learning algorithm. Cameras are mainly located around the windscreen but can also be anywhere onboard. Exceptions are the Tesla self-driving cars, in which cameras are located on top of the vehicle for the lateral shooting of the surrounding area [28].

- Global Positioning System (GPS): The vehicle uses the GPS to estimate the precise location and plan a journey. The GPS information guides the vehicle on its path to its destination.

- Other Sensors: Many other sensors facilitate the vehicle's safety and generate data from the environment for several purposes. Odometry is used to measure the wheel's displacement and speed for calculating the velocity and position of the vehicle. Velocity and acceleration data are also provided by Inertial measurement units (IMU), which additionally adjust the vehicle's speed according to the road's orientation. Furthermore, the Tyre pressure monitoring system (TPMS) reports a flat tyre condition to the control unit. A brief description of other types of sensors is listed in Table 2.5 to enlighten the reader about the kind of data collected.

Control Unit

The main control unit in modern cars and AVs is the Electric control unit (ECU). It processes collected data and initiates control decisions. Researchers reported over 50 to 70 ECUs in AVs with around 100 million lines of codes across the ECUs [57]. ECUs are grouped into several levels of functionalities. First is the Powertrain, referred to as the brain of the ECU, controlling other control modules, handling the transmission, battery charging, and emission of exhaust fumes. Secondly, the safety systems are responsible for safety controls, e.g., obstacle avoidance, active braking, and emergency stop. Thirdly the body control, e.g., door locks, air condition control, window, and side

mirror control. The last category handles intra-vehicle data communication among all devices.

- Control Area Network (CAN): A data link and physical layer communication protocol interconnect ECUs. The main advantage of using a CAN bus in an AV is maintaining compatibility between two CANs. Different new modules can be integrated into or removed from it without tampering with the whole vehicle's wiring configuration. CAN is structured in three parts: the Data link layer, high-speed and low-speed CAN physical layer. Similarly, in functionality to CAN is the FlexRay which is faster (10 times the speed of CAN) and more reliable in operation (supporting 254 data bytes instead of the 8-byte limitation of the CAN and a more secure authentication protocol). It is worth mentioning that the CAN uses challenge-response pairs for authentication between ECUs, as the encryption algorithms are never to reside on the node. Research showed that the challenge seed is crackable in a few days, depending on the size of the seeds [57].

Storage Device

It is an event data storage device like the black box concept in an aeroplane. The device stores traffic information for accountability purposes. Though storage devices are installed into some recent non-AVs for insurance purposes, the storage purposes in an AV could be diverse. Several new storage device types are proposed for more traffic-related communication (e.g., Event Data Recorder (EDR)) and decentralisation characteristics (e.g., Blockchain). According to the NAVYA safety report on the stor-

age device, the black box stores a vast amount of data, including obstacle detection raw data, position data, camera streams, vehicle status, commands, etc. Some data, e.g., status and events, are sent to external storage through NAVYALEAD for analysis. However, access to the black box is requested if more analysis is necessary. As part of NAVYA's privacy policies, camera data are stored in the vehicle. However, if the video streams are needed, they do live streaming without storing it or retrieving logs from the vehicle [32].

2.3 Generative Methods

The basic definition and preliminaries of variational autoencoders (VAE) and clustering are introduced in this Section. Their applications are fundamental to the development of DeepClean.

Variational Auto-encoder (VAE) - is an artificial neural network architecture that compresses input data into a latent space and decompresses the encoded sample to generate a similar sample to the original input data, with a cost function that can be minimised to improve the quality of the output. Minimising the cost function results in the optimisation of the evidence lower bound (ELBO) or also referred to as the variational lower bound (VLB) [60].

The major difference between VAE and regular auto-encoders (AE) is how it represents each layer as a distribution (such as the mean and variance of a Gaussian) instead of a data value. Also, the VAE performs a feed-forward neural network operation

comprising an encoder and a decoder. The encoder compresses the input data X into a vector $q(z)$, a probability distribution of z like the regular AE. Then sampling z from the distribution $q(z)$, the decoder takes in z and decompresses it to another distribution $p(z)$. An observed sample similar to the original image can be obtained from it. Sampling z from the Gaussian distribution $q(z|x)$ requires the input x, referred to as posterior predictive sampling. On the other hand, sampling z from $q(z)$ without the input data is referred to as prior predictive sampling. Several VAE models have been developed that assume various prior predictive sampling methods, such as DirVAE, GVAE [60], GVAE-Softmax [61], SBVAE-Kuma [62], SBVAE-Gamma [62], and DirVAE-Weibull [63,64]. For instance, GVAE and GVAE-Softmax samples z from a standard normal $\mathcal{N}(0,I)$, SBVAE samples using Griffiths-Engen-McClosky (GEM) procedure, DirVAE used the Dirichlet distribution assumption on the latent variable z. These methods produced more interpretable latent representations and likewise achieved better representation accuracy. This sampling adopts the normal standard Gaussian distribution to achieve good representation learning.

Clustering – remains a fundamental unsupervised learning task in computer vision and deep learning, mainly targeting grouping a similar set of data objects into the same cluster [65]. Many problems, including image analysis, have been widely studied and solved using clustering methods. The most popular methods are roughly categorised into hierarchical, centroid-based, and distribution-based clustering [66]. In hierarchical clustering, grouping is modeled based on distance measurements, and ideally, their algorithms connect nearby objects to form their clusters. Agglomerative

clustering is a well-studied hierarchical clustering problem that builds a bottom-up clustering hierarchy by taking a single cluster (containing only one data object per cluster) at a time [67]. The clustering modelling process of hierarchical is usually time-consuming and unsuitable for problems with large datasets [68]. One of the most efficient and well-studied centroid-based clustering is the K-means algorithm, which randomly chooses K samples from the input data as cluster centres at first [69]. Then each sample is allocated to its closest cluster centre based on a particular proximity measure. Updating the cluster centre is done once the clusters are formed, and lastly, repeat the steps until converging to a solution point. The optimisation problem of K-means, which aims at minimising the score function, is said to be NP-hard. Thus, methods like Lloyd's algorithm provide appropriate solutions to find a local optimum [70]. Another centroid-based clustering like kernel K-means [71], matrix factorisation, has also been used to achieve high-performance clustering tasks. Therefore, centroid-based clustering is more efficient for large-scale data than hierarchical clustering, which requires high computational power [72]. However, they cannot generate samples needed to sample the latent variable in the proposed generative model.

Distributed-based clustering, like the Gaussian mixture model (GMM), produces an efficient clustering solution similar to K-means and its variance [73]. GMM assumes that data samples are modeled from Gaussian distributions whose parameters are optimised by the Expectation Maximation algorithm. GMM generates samples from data density estimation and captures the correlation and dependence between dataset attributes because of their ability to conduct clustering analysis on good data represen-

tations instead of raw data values. These properties of GMM are the major advantages over the K-means algorithm. Hence, GMM would better suit the purpose of learning better representations of the street-view images and capable of generating samples used in the proposed generative model.

Machine learning techniques and their broader families, such as deep neural networks (DNN) and convolutional neural network (CNN), have recently achieved significant improvement in image representation [74]. Examples of deep learning-based applications for image representation are AlexNet [75], FCN [76], AE [77], VAE [77]. These applications generate better learning representation than just the dense neural networks [78]. However, many works explore combining clustering techniques with representation learning [79,80]. The categories of such applications are two folds. One is a two-stage work that runs clustering algorithms after the feature representations on the latent embeddings, and the other jointly optimises the representation learning and clustering [81]. The deep clustering architecture of this work falls under the latter based on a fully connected network. This network will allow the easy monitoring of the privacy account of the proposed private GMM. Most recent studies focus on deep clustering without considering privacy guarantees. *Yang et al.* jointly learn in a recurrent framework, deep representations using agglomerative image clustering as a forward pass and CNN representation learning as a backward pass. *Xie et al.* jointly optimised deep embeddings and clusters with KL-divergence, using stochastic gradient descent via backpropagation on the clustering to learn the mapping. *Wang et al.* formulated DTAGnet for deep learning clustering with the initiative of a graph-regularised

sparse coding for a feed-forward network and clustering task for better performance. The work of *Jiang et al.* [73] is closer to my generative modelling. It combines VAE and GMM for the encoder network and another deep neural network for the decoder network and optimises the evidence lower bound. However, their works did not implement privacy protections.

A few other studies considered privacy guarantees for the output data of their model. *Acs et al.* build on a differential privacy framework to design a private generative model. A private K-means clustering algorithm (that adds noise directly to a generative model) combines with VAE to generate privacy-preserving image data. However, their model uses a K-means algorithm incapable of generating samples and a less visual representation performance than the GMM model. Some other works also applied L_1 distance to enforce privacy, where a larger L_1 distance indicates more privacy preservation and a smaller L_1 distance means less privacy-preservation [15]. However, there is no work on a private GMM with a generative model (VAE) for street view image representations and privacy guarantees.

Table 2.1: Privacy Terminologies

Privacy Terminology	**Definition**
Privacy	The protection of identity [58]
Personal Information	Information that identifies, relates to, describes, and is capable of being associated with or could reasonably be linked, directly or indirectly, with a particular consumer or household: California Consumer Privacy Act (CCPA) [59]
Personal Data Breach	A breach of security leading to the accidental or unlawful destruction, loss, alteration, unauthorised disclosure of, or access to, personal data transmitted, stored or otherwise processed: General Data Protection Regulation (GDPR) [49]
Sensitive Information in a Dataset	Information from which we can learn something. E.g., To learn someone has a disease, to learn a user's whereabouts.
Quasi-Identifiers	Revealing attributes that can potentially identify an individual when taken together with other related or dependent attributes. E.g., Timestamp.
Privacy-Preserving Techniques	Techniques used to reduce the data disclosure risks while preserving some data utility.

Table 2.3: Data publication privacy threat [51, 146]

Privacy threat on AV dataset	**Description**
Re-identification	A reverse process of de-identification. The real identities of subjects are obtained from a set of hidden identities in a dataset. Examples of explicit identifiers are Vehicle Identification Number, Name, IP Address, Image, and Video.
Attribute disclosure	Revealing quasi-identifiers and sensitive attributes that give enough information or knowledge to identify a subject. Examples are Spatial-temporal attributes in a trajectory dataset.
Correlation	Establishing a relationship between the dataset and any other piece of side information. An example is a temporal correlation between two datasets.
Inference	An analyst gains accurate knowledge from an individual or group dataset by a data mining technique for the wrong reasons. For example, an illegitimate analysis of trajectories for places visited to gain secondary information like the user's religious status.
Linkability	Linking two or more events or activities of the AV together, for example, linking accidentally captured objects and images with actual locations or an identity.
Observability	Revealing the subject's activities, for example, correlating an event with a subject.

Table 2.5: Vehicle Sensor generated data.

Category	**Name**	**Description**
Time	Time	Current time
Mode Change Info	Mode	Current driving mode
	ModeIn	Previous driving mode
Driver Input	APS	Accelerator pedal position value
	BPS	Brake pedal position value
	SAS	Steering angle sensor value
	Gear	Transmission gear(P,R,N,D)
	OBD2Spd	OBD2 based vehicle speed
	PushBrake	Push Brake (on/off)
	Steering Torque	Steering torque value
Sensors	Lidar	Obstacle detection raw input
	Radar	Ranging obstacle detection raw input
	Odometry	Displacement and speed of wheel
	TPMS	Tyre condition (normal 0/ flat 1)
Vehicle Status	Longitude	GPS Longitude
	Latitude	GPS Latitude
	Wheelspd	Vehicle rear-wheel average speed
	VelCMD	Target Speed for autonomous driving
	VelCurr	Current speed of the vehicle
Camera Data	Video	Obstacle recognition raw stream
	Image	Obstacle detection image
V2V Data	BSM	Road conditions, braking intention, turning intention, road closure, road condition

Chapter 3

Literature review

3.1 Introduction

This chapter presents an in-depth analysis of the literature with methods used in previous research on AV data privacy and points at the research gaps. It is divided into five sections, showing a detailed literature review of privacy-preserving methods and AV privacy challenges. Section 3.2 explains various privacy-preserving methods to tackle privacy weaknesses in information systems. Section 3.3 looks further into privacy-preserving methods in data publishing with a subsection that explains several anonymisation techniques and their applications to private data. Section 3.4 explains geo-localisation techniques to build on mounting location attacks on AV camera data. Sections 3.5 and 3.6 elaborate on the data privacy challenges of AV sensor data and connected vehicles. They point out issues addressed by previous works and highlight gaps that could be addressed. Section 3.7 summarises the relevant works, gaps and the contribution of this research.

3.2 Privacy-Preserving Methods

Machine learning techniques are widely utilised and applied with traditional privacy techniques like K-anonymity and differential privacy to solve privacy problems in data

mining, publishing, and storage [82]. A fundamental part of machine learning is clustering, which involves grouping a set of similar objects in clusters [52]. Its application in computer vision tasks, e.g., object detection, face recognition, and image analysis, has been widely studied and achieved efficient performances. Usually, efficient clustering algorithms are justified by how well they can represent data, generally achieved by solving an optimisation problem. However, the more complex the features in an image or video data, the more difficult it becomes to generate a well-structured representation of the data using many existing clustering algorithms [78]. Recent works focused on deep learning-based image clustering approaches for feature representations in an unsupervised setting, which have shown to be more efficient than supervised settings. For example, the studies in [78, 83, 84] performed the data generation process using an unsupervised approach, aiming at learning a joint distribution of images in different domains by using images from the marginal distribution in individual domains.

Yang et al. represented images using agglomerative clustering and activations of convolutional neural networks [84]. *Hsu et al.* also proposed a clustering convolutional neural network to capture better the salient part of an image without the need to provide any bounding boxes in the training stage for a better representation [78]. *Wang et al.* combined a Sparse coding base pipeline into deep learning for clustering, achieving an extremely efficient inference process and high scalability of large-scale data [83]. However, these methods are only efficient on images with fewer features like the MNIST dataset and do not consider privacy in the image generation process [80]. To this end, the image translation performance of VAE and the GAN models has been

remarkable recently. *Liu et al.* proposed an unsupervised image-to-image translation framework, UNIT, based on GAN and VAE [85]. These adversarial training objectives interact with a weight-sharing constraint, enforcing a shared latent space to generate corresponding images in two domains. At the same time, VAE relates translated images with images in the respective domain, presenting high-quality image translation results for street-view images and videos. This thesis considered works that design generative models with clustering tasks with the data generation power of VAE outperforming GAN-based models and considering privacy protection.

Recently, *Xiong et al.* were the first to address privacy concerns of auto-driving images and videos. The auto-driving generation neural network (ADGAN I) uses UNIT to generate data and applies noise directly to the original image to produce the synthetic samples [15]. The noise addition approach degrades the quality of the synthetic samples because of the mirage of information contained in the original dataset. This method may be suitable for smaller images with less information, not street view images with complex information. ADGAN II combines GAN with VAE to better represent street view images [14]. With the use of VAE in ADGAN II, the synthetic samples can now be produced by a latent vector without any original data, which makes ADGAN II more flexible for real applications like the street view image. Generally speaking, GAN-based models may lose perpetual accuracy due to the model collapse property of GAN.

For this reason, several methods, such as Mean Square Error, Peak Signal-to-noise Ratio, and Structure Similarity Index Measurement, are used to access high perceptual

accuracy. However, the more complex the images, the less effective the GAN-based models [86]. A few other methods proposed a stronger data generation technique utilising the data generative power and useful basic generative structures of VAE with deep neural networks for clustering tasks. *Acs et al.* divided data into clusters using a differentially private clustering approach, giving each cluster a separate generative neural network such as VAE to train on their cluster using differentially private gradient descent [87]. The partitioning of data into general clusters led to more accurate synthetic samples than just training the whole dataset as a single model. A more powerful clustering framework was proposed in [73] combining VAE and a Gaussian mixture model and maximising the evidence lower bound using Stochastic gradient variational Bayes estimator and the reparameterisation trick. The proposed method of this thesis adopts the data generation technique in [73] and optimises GMM for clustering tasks to generate complex image features.

Some approaches to privacy preservation ensure selectively distorting data features to balance privacy/utility trade-offs. In response, *Chong* proposed a generative adversarial network that reduces privacy risks by removing location-relevant information [64], e.g., background buildings, from the camera data, before being used for analysis. The location-relevant information in the camera data was analysed and reported as a threat to privacy when providing the data for analysis. Location-relevant information in the camera data highlights privacy risks. Trajectories of a vehicle could be formed or traced by extracting the location hints from image data and matching them with reference data to geolocate them. However, camera data may also contain

other quasi-identifiers (QIDs such as the human face and vehicle plate number) besides location-related ones, putting users' privacy at risk.

Only two studies (previously explained in this section [14, 15]) addressed location inference threats for AV-generated camera data to the best of my knowledge. Their solution to the problem involves using VAE and GAN-based models to generate privacy-preserving datasets. Using GAN in their approach has two practical limitations. One is that the privacy achieved by the discriminative distance measurement cannot guarantee the location privacy of the image data. Secondly, a robust geo-localisation tool can exploit the discriminative distance value of the original and distorted images to estimate the geolocation of the target image. Therefore, this thesis proposes a strong attack model to show the inefficiency of the existing literature and then suggests a method for separating different parts of an object to add more noise without affecting much underlying structure of the non-private object parts.

3.3 Privacy Preserving Data Publishing

Statistical agencies release various data types in a format ready to be processed for promoting advanced research, collaborative learning, and analytics [88]. Often released data reflect areas in medical health, social networks, census, and surveys [89, 90]. Recently, datasets generated by AVs during experiments have been published and made accessible by organisations like Google [30], Berkeley [91], and Lyft [92].

The released data aids the research and development of higher functionalities and in-

telligence such as 3-dimensional perception and prediction, global localisation, object detection and recognition in AVs [39]. Generally speaking, released databases contain attributes that could be categorised under these descriptions: (1) attributes that contain identities, "referred to as identifiers, e.g., identification number and social security number"; (2) attributes that are linkable with other attributes to identify a person, "referred to as quasi-identifiers, e.g., zip code and sex"; and (3) attributes that provide new information about a person, "referred to as sensitive attributes, e.g., disease" [93]. Attributes in AV dataset also fit these descriptions; for instance, an image explicitly identifies a person, and an accidentally captured object in an image, such as a passing vehicle's plate number, may be referenced as a quasi-identifier in a re-identification process. Something sensitive could be learned from released trajectories, e.g., the vehicle's whereabouts, most pick-up locations, and details of the journey [94].

The application of privacy-preserving techniques on the dataset reduces the chances of re-identification. However, there have been reports of high-profile, successful re-identification attempts on publicly released data, e.g., Netflix data [94] which *Narayanan* and *Shmatikov* matched some users' records in the dataset with another publicly available internet movie database. American online, New York taxi and Limousine commission datasets were also an example of successful attempts [95].

Various privacy technical controls are applied to the data before release, limiting what an observer could learn. Disclosure level is often evaluated based on how many sensitive attributes could be learned, given the capability of an adversary (i.e., how much background knowledge is known). However, in statistical disclosure control, disclo-

sure level is evaluated by how much inference is possible on a group [96]. In this case, disclosure levels are evaluated on a group of a given camera dataset.

Utility degradation or unavailability of data for public use are the challenges observed when attempting to release a dataset with high privacy restrictions. Many works tried to improve on balancing the trade-off between privacy and utility to enhance privacy-preserving data publication [97]. The requirement to evaluate utility preservation observed from the existing works is to compare the model's statistical properties of the released dataset to the original dataset. Then, some important properties could be retained depending on the data's use case. Therefore, in choosing a technique to achieve the objective, this study will also consider various data properties of the released dataset and the possible threats to the data.

3.3.1 Anonymisation Technique

Over the years, anonymisation has been the prominent data privacy-preserving approach, intending to reduce disclosure risk while retaining data analysis benefits [98]. K-anonymity provides a simple method to alter identifiers by generalising or suppressing them [99]. The model's strength relies on the general-purpose data usefulness offered through partitioning strategies to produce classes of indistinguishable records. It efficiently protects online social network privacy and databases [100], especially protecting social network users' identities and hiding social links [101]. A significant limitation is the lack of attribute disclosure protection because it assumes that only a few attributes could be used as quasi-identifiers when carrying out a linkability at-

tack [102]. A dataset may have many attributes with a few related quasi-identifiers. The well-known case of the re-identification of the Governor of Massachusetts from a k-anonymized public dataset where attributes like social security numbers and names have been removed is an example of the limitation of K-anonymity [103]. His identity was known by linking attributes like zip code, gender, and date of birth from the dataset.

To solve this problem, an improved variation of K-anonymity known as l-diversity ensures that each equivalence class's distribution of sensitive attributes has at least l well-represented values [104]. Though a few methods, such as distinct, entropy, and recursive, achieved diversity among attributes, the methods still suffer from similarity and skewness attacks [105]. T-closeness is another improved variation that solves similarity attack problems by ensuring the distance between two distributions of sensitive attributes should be no more than a threshold t [106]. This approach defines a stronger attribute privacy notion and offers improved analytical value. *Soria-Comas* [105] showed that t-closeness could satisfy the privacy guarantee of differential privacy and outperform it in retaining data utility. Since the advancements in machine learning techniques, adversaries leverage their power to exploit the weaknesses of k-anonymity techniques and their extensions through inference [107]. Due to the privacy challenges of big data, the unlimited capabilities of the adversaries, and the challenges in producing valuable synthetic data, K-anonymity is inefficient on high-dimensional datasets such as images. It does not sufficiently provide privacy guarantees and significantly degrades the quality of the data. Thus, many works focused on a more robust privacy

method, such as differential privacy.

Differential privacy offers a mathematically proven privacy guarantee without limiting its assumptions of the adversary's capability [108]. It builds a stronger immunity to inference attacks. It is undoubtedly widely accepted by industries and organisations dealing with data processing. Differential privacy was originally applied to interactive data in database querying, where the utility of query results is a priority [108]. Many of its applications on non-interactive databases also exist. However, this part reviews the essential features and factors that facilitate the optimisation of the technique regarding privacy notion and the utility of non-interactive data access. Intuitively, differential privacy injects random noises into the input dataset to produce an output that does not reveal much information about any input data record [109]. Several approaches achieve distortion of the dataset. The most popular are the Laplace mechanism for generating numerical outputs and the exponential mechanism for generating categorical output [109]. The distortion of the dataset achieved by the mechanism has been criticised for significantly affecting utility in mining and analysis tasks [54]. Mainly, the noise injected sometimes outweighs the data signal, which results in data utility loss. Adding Gaussian noise instead of the Laplace mechanism has been mostly utilised in practice because of its easier practical understanding and corrections to the privacy mechanisms [110].

Several relaxations, such as local differential privacy [111] and individual differential privacy [54], attempt to improve the functions and ensure that the distortion does not render the data useless. Applications of differential privacy with conventional ma-

chine learning and data mining operations are well studied. Conventional classification data models such as decision tree classifiers [112] and support vector machine classifiers [113] achieve better privacy and utility preservation. Nevertheless, with the advancement in sensor technology, leading to the generation and processing of huge sensor data, deep learning techniques have outperformed conventional learning techniques [114]. The deep learning techniques can produce better models as the dataset grows. The application of differential privacy in deep learning models has also been extensively studied. The result has led to some organisations releasing their experimental data for analysis (for example, to be used for collaborative learning) [115]. However, machine learning approaches are vulnerable to model invasion attacks that exploit confidence information, such as matching people's faces with a facial recognition database [116]. Many studies evaluate the potential of differential privacy to defend against machine learning-based inference attacks [107]. *Dpm et al.* attacked a privacy-preserving collaborative deep learning protected with distributed DP, using a GAN-based approach to show that participants' training sets are not protected in a decentralised approach.

Privacy-preserving frameworks guide the application of differential privacy with optimisation problems in several learning models such as Neural networks [115], Bayesian networks [117], Logistic regression [118], Support vector machine [113], and association classification [119] for efficiency. Recently, noisy gradient descents have been used to obtain optimal error for minimising Lipschitz convex functions over the l_2-bounded set. Google used noisy Stochastic gradient descent on the MNIST dataset to

achieve good private model accuracy. Also, IBM privacy showed the implementation of noisy variants of logistic regression and Naïve Bayes on the Adult dataset to generate good private models. Tailored noisy variants of learning models, e.g., Histograms and Bayesian networks, were optimised to improve data usability in producing synthetic datasets (with the 2016 and 2017 SFFD's call for service data and Colorado PUMS training dataset) in a NIST competition. These algorithms have been used on different types and structures of datasets to achieve optimum performance based on various objectives. Depending on the dataset type, linear algorithms may perform better than non-linear ones and vice-versa. Nonetheless, the creative implementation of the privacy-preserving framework must be evaluated to suit the AV data analytics or storage objectives.

Data synthetization defines a more solid framework for releasing huge volumes of privacy-preserving data [120]. It involves modelling the original data statistics and then generating a new set of data from the distribution while it preserves some important statistical properties of the original dataset [120]. From the evaluation of synthetic data by *Loannis et al.* [121], if the aim is to generate a very detailed multivariate distribution, then a large part or all dataset from the original dataset is likely to remain, which may likely result in disclosure risk. Although, generating a lower detailed multivariate distribution preserves confidentiality because new datasets are produced. Some data relationships may be lost, rendering the data useless for certain analyses. Synthetization frameworks that define the analytical use case and disclosure risk level can derive efficient results (when balancing users' needs against disclosure risk). Since syntheti-

sation techniques present several flaws, some existing works utilised their strength with other privacy-preserving techniques to achieve a better trade-off. *Gursor et al.* [122] synthesised privacy-preserving traces by enhancing the plausibility of synthetic traces with social networks. In the data privacy-preserving scheme using a generalised linear model, Lee [118] replaced high disclosure risk values with synthetic data sets instead of adding excessive noise. The approach preserves data relationships in the original data for better analytical purposes.

3.4 Geo-localisation Techniques

AVs are a major source of street view images alongside other sources such as social media and photographer's databases [123]. AV images are essential to developments in applications such as smart cities, smart transportation, insurance liability claims, and real-time and offline image analysis for AV models [124–126]. However, geo-localisation approaches in an attack setting potentially threaten the privacy of AV-generated street-view image data through its development in object, scene detection and matching techniques [127]. *Xiong et al.* show that objects in these images contain contextual information cues about geolocation. Detecting and matching the information with high probability may give away location details. Prior studies proposed geo-localisation approaches for location inference attacks and considered several implementation conditions. Both conditions depend on the data the approach has been developed for and the target environment.

Image-based methods and data of modalities are prevalent input data types for the

methods [128]. Their target environment focuses on three main areas: city scale, global scale, and landscape [129]. Since the AVs mainly capture data in cities showing objects with distinct structures such as buildings, road signs, and surrounding vegetation, this thesis focuses on city-scale geo-localisation in an urban environment. In recent works, city-scale techniques showed more geolocation estimation than the other areas [130]. However, none of the works considered exploiting privacy-preserved data as the query for testing the technique. As in this case, the technique is improved to geo-localise noisy GAN-based generated data with higher performance than the prior works.

The other area, like global scale, focuses on a large-scale geo-localisation of the earth, presenting a greater challenge of localisation estimation than the city-scale [131]. Techniques proposed in recent studies did not perform well on a global target. On the other hand, landscapes focus on geo-localising nature, such as mountains, popular buildings, oceans, deserts, and foliage [132]. This thesis focuses mainly on techniques developed on image-based for a city-scale target. It highlights the image retrieval performance in the context of a location inference attack on AV data.

Early techniques use appearance-based matching to capture important appearance information [133]. *Kosecka et al.* used a gradient orientation histogram to extract information. *Hayet et al.* extracted planar quadrangular landmarks from images and obtained invariant representations for a principal component analysis (PCA) learning stage using homography rectification. However, the techniques do not sufficiently handle occlusions and clutter [134]. Recent methods such as feature-based matching robustly solve image geo-localisation problems and handle feature generation in object

recognition with constant adaptation to several transformations such as scaling, rotation, translation, colour changes, distortion, and perspective projection [16]. This technical robustness makes them suitable for solving geo-localisation problems of a changing environment and invariance to image quality. Examples of feature-based image matching methods include Scale-invariant Feature Transformation (SIFT), Speeded Up Robust Features (SURF), constrained condition SURF, Robust Independent Elementary Features (BRIEF), Oriented FAST rotated BRIEF (ORB), etc.

Lowe's SIFT [135] for image feature detection has been implemented in various domains such as remote sensing [131] and geo-localisation. Although SIFT achieved lower efficiency in terms of computational complexity than the other feature-based methods, it performs very efficiently in object recognition applications. It detects more features from distorted and noisy image data than the other techniques [133].

Schindler used vocabulary trees to organise SIFT features that are most informative about each location and a voting scheme to find the best matching reference image [12]. The data used in their work are 20km urban street imagery from a vehicle's camera. The method used on the data improves performance by increasing the number of database images by a factor of 10. It also improves storage and time complexity by assigning a node to each descriptor without requiring direct access to each database feature at search time. Improved works focused on efficiently organising the data in the database and choosing an efficient image-matching technique. *Zamir* evaluated their technique on 102k Google Street View images and outperformed previous works by 10% [12]. Their generalised minimum clique graphs for geo-localisation with novel

multiple nearest neighbour feature matching methods present additional improvement in feature matching by extending the definition of a node to a cluster node. The global context of the image, such as GIST, colour histogram and image geo-tag, were implemented and shown to improve the performance. Geo-tag yielded the best overall results and voting scheme similar to [13] to find the reference image that most strongly matched the query. *Zemene et al.* proposed a technique that offers several advantages over the existing approaches [11]. The first is to dynamically select improving numbers of nearest neighbour features using dominant set clustering. The second part improves the time efficiency better than the prior technique. Constraint dominant set clustering used in the study bypasses heuristic approaches to voting by computing the highest membership score as the final best-matching reference image. Chapter 4 of this thesis employs the image matching technique in [11] to improve the discriminative power of noisy NN features and adapt the technique to improve geo-localising distorted images.

Geo-localisation solutions with convolutional neural networks achieved good accuracy on street scene images [136]. They also localise many photos and predict the geographical attributes for image retrievals. Although CNN approaches are well-suited for this problem, they do not perform well on distorted images because they learn and predict the features present in the image and cannot efficiently match the images using their nearest neighbour estimations. *Karami et al.* show that SIFT robustly handles image matching against various distortions [137]. Therefore, this thesis adopts the local descriptor of SIFT to extract noisy objects from the GAN-based privacy-preserving dataset. More so, we can flexibly use the calculation from the discriminative distance

measurements of a noisy object and its NNs to better estimate an approximate matching image.

3.5 AV Sensor Data Privacy

AV sensors data, such as location, camera, and distance data, e.g., Lidar and Radar, are identified as sensitive data. They could also be considered quasi-identifiers to link with other data in breaching data privacy. Having unauthorised access to the data may initiate a privacy attack. Hence, privacy constraints pose significant issues that hinder access to the data for any use cases (e.g., for AV navigation localisation application [113]). The rest of the section discusses how data are a potential risk to traditional privacy threats such as re-identification, attribute disclosure, correlation, and inference. (Table 3.1) shows the effectiveness of several privacy-preserving techniques on high-dimensional data and their potential privacy threat.

3.5.1 AV Location Data Privacy

Many AVs use a localisation application, e.g., Geographical Positioning System (GPS), to guide the vehicle's movement with routing information and services involving precise location updates such as alternative routing, road condition info, and traffic condition info [104]. The location information collected for the service offered consists of vehicle events. The events are represented as a series of geographical positions with timestamps. For example, when a vehicle communicates anonymously with a Location-Based Service (LBS), the LBS holds some information about a de-identified

vehicle's starting and destination positions, routing, and time information. Publishing the location data set in which identities are pseudonymised to ensure de-identification is not enough to protect privacy [105]. An adversary could infer the user's attributes like home address, office address, medical condition, gender, parking positions, point of interest, journey pattern, and more. Vehicle users' privacy may be at risk, as serious as re-identification, if the attributes link to other data sources.

To ensure the privacy of location data, many works in intelligent transportations [101], smart cities, connected vehicles, mobile devices, and social networks employed suitable location privacy-preserving methods (LPPM) to distort the correlation among quasi-identifies (QIDs) and unlink relationships between events [122, 138]. Several LPPM falls under these broad categories: K-anonymity, expected distance error, and differential privacy [139, 140]. Depending on what type of application generated the trajectories, QID attributes may vary, and applying LPPM must consider this variation. *Huo et al.* argues in a trajectory privacy-preserving article that different trajectories may have different quasi-identifiers apart from the main QIDs (e.g., spatial and temporal attributes) [138]. Also, some other works argue that QIDs are more complex to define in trajectory databases [141]. Specifically, about vehicles, *Huo* showed that parking locations are potential QIDs that an adversary can utilise to link a vehicle user [138]. Another practical example showed that timestamps are powerful QID when continuously releasing trajectories [142]. An accidental correlation of timestamps may occur if trajectories are released often. Apart from quasi-identifiers in trajectories, the puzzle of adversarial knowledge is a major challenge when considering LPPM for data

release. An example is that an adversary may know or formulate some background knowledge about a public/private AV. For example, someone may know the time and location of an AV that operates publicly. With this knowledge, the rough trajectories may be computed from a set of anonymised ones achieved through location privacy techniques like deletion, generalisation, error addition, noise addition, etc.

From the differential privacy perspective, the geo-indistinguishability technique robustly solves location data privacy suitable for publication, considering the uncertainties of adversarial capabilities [143]. The recorded locations in the sample dataset are calculated as the probability of the maximum difference from their real locations. One issue with this technique is the over-protected problem in a real-life application which reduces the quality of data use [144]. Several optimal approaches attempted to maximise data quality by solving linear optimisation problems. However, intuitively, the semantic relationships between the dataset are degraded. The degradation may affect utility in some AV use cases of the dataset, e.g., developing a localisation application for AVs. The sophistication of location attacks like inference attacks, position correlation attacks (for instance, correlating trajectories with location information in image data), and temporal correlation attacks encourage stricter location privacy notions, which otherwise diminishes data quality.

3.5.2 Distance Data (Lidar and Radar) Privacy

Distance data from Light detection ranging (Lidar) and Radio wave detection (Radar) are represented in a continuous data form. Their privacy can be preserved using any

privacy-preserving methods applied to location data. However, there are assumptions that distance data are possible useful QIDs [50]. Since sensors detect road obstacles around the vehicle, each data recorded implies the distance between the vehicle and the obstacle at a time. In essence, the recorded distance data indicates the presence of an obstacle, e.g., other vehicles, pedestrians, buildings, etc. Given the data, an adversary may roughly compute the vehicle's movement details, e.g., consecutively, close distance information may suggest traffic on the road or the type of road, and far distance information may suggest free-flowing traffic. Linking the distance data with speed, camera, and location data could infer the vehicle's movement more accurately. An observation by Bloom shows that Lidar data can be combined with other sources to count how many people are at a protest [9]. Their suggested technical solution is integrating privacy-enhancing technology (PET) into a smart city by introducing citizens' privacy, which is the whole community's right to privacy. Thus, given the detection and tracking capabilities of the sensors, different privacy concerns may yet be unveiled.

3.5.3 Camera Data Privacy

The processing of camera data consisting of images, videos, and contents sparks the most intense legal debates in AV privacy [94]. The data generated by the camera sensors contains images of road vehicles (their plate number and model), pedestrian faces, buildings, road signs, and anything within the view of the sensor, which is most sensitive and attracts privacy threats such as location inference and identification of

objects. However, existing privacy-preserving methods have been used previously to protect the privacy of camera data. A list of camera obfuscation methods was discussed in [145], which include blurring, blocking, pixelating, inpainting, silhouette, and morphing. However, the methods are perturbative and ensure noise addition to make the data obscured and adding too much noise renders the dataset less useful for training an operational AV model [7].

Moreover, existing techniques are effective on small images, e.g., a person's image and numbers. However, images with many relevant objects and analytical importance, such as street view images, will require a more efficient privacy-preserving technique. The important elements preserved when producing camera data for AV analytics may include road entities such as road markings, pedestrians, cyclists, and other vehicles. This is essential information to model AV navigation from one location to another. Any other entities in the camera data may be categorised as over-needed information, e.g., accidentally captured vehicle plate numbers, road information, name of places, popular buildings, structures etc. [146]. This information could be used in an attack to put location privacy of the vehicle at risk.

Objects in images must be selectively perturbed to balance privacy and utility trade-offs to protect AV camera data. More so, current applications must be able to address the current threat that cannot be sufficiently handled by the privacy techniques on camera data mentioned above. In response, *Chong* proposed a generative adversarial network that reduces privacy risks by removing location-relevant information [147], e.g., background buildings, from the camera data before being used for analysis. The location-

relevant information in camera data was analysed and reported as a threat to privacy when providing the data for mining. The trajectories of a vehicle could be formed or traced by linking images of buildings, structures, and road entities in a video or set of images with time information and map application. However, camera data may also contain other QIDs besides location-related ones, which could put users' privacy at risk. Thus, this study evaluates the privacy-preserving techniques for camera data to address current threats and utility satisfaction of the data for storage and processing.

3.6 Connected Vehicle Privacy Challenges

Connected vehicles present one of the technical data access points identified by *MC Kinsey* [36]. Data sharing between the connected devices undoubtedly improves vehicle safety and convenience [36]. However, the more connectivity of road devices to share data, the more privacy risk introduced [47].

One major area of focus is the Vehicle to Vehicle (V2V) communication with the basic safety messages (BSM) generated. The BSM beacons generated contain safety-related and location-related information such as position, speed, brake intentions, road closures, etc., to warn nearby vehicles of the rapidly changing surrounding areas.

The US Department of Transportation researched V2V and made efforts to make the basic safety message (BSM) contain no identity, requiring no authentication and identity hiding techniques when sharing information with nearby vehicles [41]. The approach is developed to address privacy concerns in V2V communication and avoid

using cryptographic techniques for privacy preservation. Some work authenticated and anonymized V2V communication beacons using cryptographic encryption and pseudonyms [36,43,44,148,149]. These methods assume that an adversary may intercept the beacons and must be kept safe. However, using encryption and pseudonyms only partially solves privacy issues in V2V. It introduces some new vulnerabilities that an adversary could exploit. For instance, encrypted beacons are analysed in data mining techniques to associate entities with a user. Zone changing by vehicles during a journey increases the chance of linking pseudonyms with a vehicle. *Burmester* [150] shows how Bayesian traffic analysis with prior information gathering was used to link changing pseudonyms with vehicle users when exiting a silent zone.

Vehicular communication beacons collected by an adversary by passively sniffing the network may not identify a subject. Still, they could be used with other vehicle data from sources like published AV data to carry out correlation or link attacks [57]. With the vast data sources and the available information-gathering access points, vehicle users' activities may be inferable by an adversary utilising advanced machine learning techniques.

3.7 Discussion on Related Works

Many pieces of literature have discussed sensitive AV data and the potential privacy threats to the data (as summarised in Table 3.1). From this table and Table 3.3 showing the effectiveness of privacy-preserving techniques, inference and correlation are serious threats to the privacy of the data and more serious when given access to multiple

AV data sources. As the table suggests, location inference threats on camera data, trajectories, and distance data were comprehensively reviewed in the past. At the same time, only two studies addressed location inference threats on AV-generated camera data. However, their solution to the problem involves using VAE and GAN-based models to generate privacy-preserving datasets.

Nevertheless, the application of privacy techniques on GAN-based applications significantly impacts the underlying non-private objects in street view image data, thus, rendering the synthetic data too noisy and unbalanced in terms of privacy-utility trade-off. To the best of my knowledge, this thesis is the first to combine VAE and differentially private GMM clustering to learn the complex structure of street view images and preserve the privacy of sensitive objects while preserving the underlying visual representations of non-private objects.

The location-inference threats on camera data from multiple data sources have been considered in this thesis. So, several data sources of the AV will be regarded as background knowledge when testing the privacy guarantee of the proposed technique. Differential privacy has been adopted to prove the statistical immunity of the proposed technique to inference attacks and correlation attacks amidst the possibility of unlimited adversarial capabilities.

Table 3.1: Summary of existing studies on high-dimensional Image representation with privacy preservation.

Existing Studies on PPTs for CAV camera data	**Machine Learning Technique**	**PPT**	**Application Description**	**Contribution**	**Research Gap**
[7]	U-Net + GAN	L_1 distance	Proposed ADGAN-I for protecting auto-driving camera data against location inference attack	Achieved privacy protection while maintaining high image quality	21% privacy susceptible to multi-KNN
[14]	VAE + GAN	L_1 distance	Proposed ADGAN-II for protecting auto-driving camera data against location inference attack	Achieved better privacy protection but less image quality than ADGAN-I	4% - 5% privacy susceptible to multi-KNN
[87]	k-means clustering + VAE	DP	Proposed a technique for privately releasing generative model	Combined kernel k-means clustering with random Fourier features to cluster high dimensional large datasets with strong privacy guarantees effectively	Cannot generate image samples due to the use of k-means clustering
[78, 83, 84]	Clustering, Deep Neural Network	No Privacy	Proposed deep learning-based image clustering	Achieved joint clustering and representation learning	Privacy guarantees are not considered in their implementation
[73]	VAE + GMM	No Privacy	Proposed variational deep embedding	Generates highly realistic samples for any specified clusters	Privacy guarantees are not considered in their implementation
[60, 61, 63, 64, 151]	VAE	No Privacy	Proposed several types of VAE	Tackle the model collapse issue of VAE to generate superior image representation in terms of qualitative and quantitative performances	Privacy guarantees are not considered in their implementation
[152–154]	GMM	DP	Proposed an algorithm for learning the parameters of well-separated GMM with DP	Balanced algorithm complexity, matching that of the corresponding non-private algorithm	Their implementation of DP on GMM models does not consider its application on auto-driving camera data
DeepClean	VAE + GMM	DP	Proposed a differentially private GMM with VAE for image representation	To achieve a better privacy guarantee compared to ADGAN-II while maintaining high image quality of non-private samples compared to ADGAN-I	my application will generate samples due to the use of GMM and attain less than 4% privacy susceptibility to multi-KNN

Table 3.3: Effectiveness of several privacy-preserving techniques on high-dimensional data.

Privacy-preserving techniques	**Efficiency of high-dimensional data**	**Application of the technique to CAV data type**	**Potential privacy threat**
Generalisation-based, e.g., K-anonymity, l-diversity, t-closeness	Low	Location data, infotainment data	Attribute disclosure, correlation, inference
Noise-based, e.g., differential privacy	High	Location data, infotainment data, distance data, camera data	correlation, inference
Synthetization	High	Location data, infotainment data, distance data, camera data	correlation, inference

Chapter 4

Location Inference Attack on AV Camera Data

4.1 Introduction

This Chapter presents a holistic assessment of location privacy threats on AV camera data and analyses AV privacy threats by building a threat matrix to capture privacy risks and highlight the potential privacy impacts on the subjects. Some security and privacy risk assumptions are justified in this section before designing the location inference attack on the accessed dataset. Security and Privacy Frameworks such as STRIDE [47] and LINDDUN [50] are employed to build the threat matrix specific to AV systems. Then, an experiment-based scenario (a geo-localisation attack) is performed to identify potential vulnerability conditions in AV data anonymisation techniques. Finally, the links between the geo-localisation attack and the location privacy breach are explained by associating the results obtained in Sections 4.4 & 4.5 with a privacy attack surface.

4.2 AV Security Threat Modeling using STRIDE Framework

The STRIDE threat modelling process consists of several steps to exploit design vulnerabilities in a system and targets appropriate protection strategies [47]. This modelling justifies the assumptions made in this thesis about the potential adversarial attacks to access AV camera data. The model processes are as follows:

4.2.1 Use Cases Definition

Identifying key functionalities is important within the modelling scope. For modelling, let's consider AVs used for private and public (e.g., taxis and commercial) services operating in urban areas, with standard functional components, sensors, control systems and applications (discussed in Chapter 2).

4.2.2 A List of External Dependencies

The definition of several applications that AV systems depend on is shown in this section. Figure 4.1 shows an overview of external dependencies on the system. Identifying several onboard applications such as location base systems for GPS, entertainment applications, other OBD service providers, Vehicle to Vehicle (V2V), and Vehicle to everything (V2X) applications.

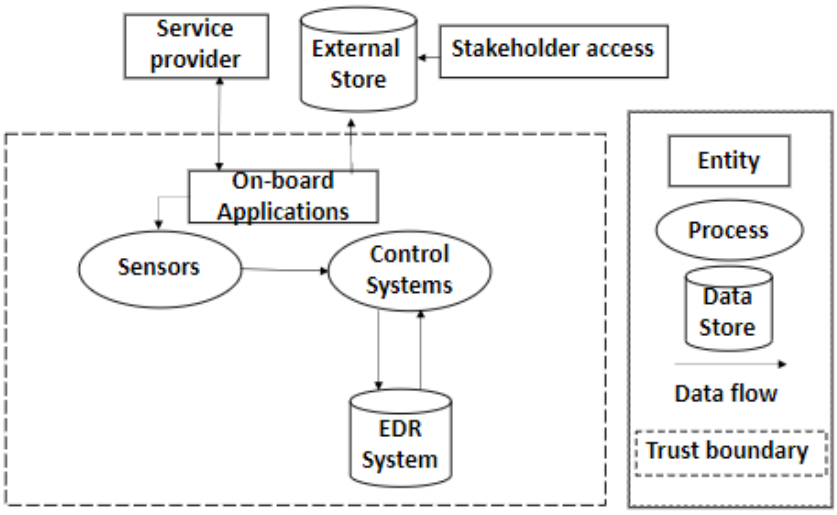


Figure 4.1: Data flow diagram of an AV system

4.2.3 Security Assumptions

Various implicit assumptions about the security protection already in place are stated in this section. Figure 4.1 shows that the main control system architecture receives data from the onboard sensors, and system communication is strongly authenticated. Also, the in-vehicle storage system, EDR, works like a blockchain system, providing untampered evidence data and implementing strong access control across the systems. These assumptions about the security level of AV internal systems suggest strong security architecture and protections that only a sophisticated adversary with strong attacking systems could potentially breach. Otherwise, without strong security protections, all other adversaries with lower capabilities in attacking tools and systems would gain unauthorised access to its communication system. Such access may include tampering with the sensor control to mislead the AV, gaining unauthorised access to the control systems for remote access of the AV, and tampering with the storage system for denial of service and repudiation attacks. The security assumptions of the external entities are discussed next regarding the AV system's architecture in this study [126]. It points out that the entities provide their security protection to communicate and access data from external entities.

4.2.4 External Security Details

Explaining the security implications of some external dependencies on the AV system points out residual threats to the AV system. The out-vehicle systems providers must ensure their data stores and flow security protection. For instance, communication

through remote data access ports must be authenticated via encryption. Physically accessing data by stakeholders (e.g., mechanics) from the AV system through the OBD port must be authenticated. Communication between onboard applications and internal entities must be done through a dedicated wireless secure communication channel. Threats to these external entities could in-directly impact the vehicle's security. For instance, a rogue or hacked vehicle in V2V settings could send wrong traffic information to another AV on the road. An infiltrated external application with the right authentication keys could access the AV's data store. This Chapter mainly focuses on the potential unauthorised access points to the AV data store through external data stores, e.g., cloud.

4.2.5 Security Threat Types

Table 4.1 defines the threat types as defined by the STRIDE threat taxonomy. Threats such as tampering, repudiation, information disclosure, denial of service and elevation of privilege (against several standard security properties such as confidentiality, integrity, availability, authentication, authorisation, and non-repudiation) are analysed.

Data tampering in an AV system could involve a hacker accessing and modifying files in the data store (DS). A hacker could tamper indirectly by using a script exploit to gain unauthorised access to an external or internal DS. For instance, an adversary masquerades as a stakeholder or service provider to access the Event Data Recorder (EDR) data. A data flow (DF) could also be intercepted and tempered to send a rogue message to confuse the AV's control. Likewise, the confidentiality of the subject could be

attacked through unauthorised access to the DS elements. For instance, location details will be learned from images/video data, the user's driving habits can be inferred, and secret network communication messages can be retrieved. An AV data flow could attract threats such as denial of service and elevation privileges by hacking the system's wireless network. Denial of service could be possible on the DS by preventing authorised accesses from the DS. Unauthorised privileged access could also breach the security of the processes by maliciously controlling the AV's sensors and systems.

Table 4.1: Security Threat types and the DFD elements they attack. E represents an Entity, DF is a Data Flow, DS is a Data Store, and P is a Process.

Security Property	**Security Threat Category**	**E**	**DF**	**DS**	**P**
Integrity	Tampering		✓	✓	
Confidentiality	Information disclosure	✓	✓	✓	✓
Availability	Denial of service		✓	✓	✓
Authorisation	Elevation of privilege		✓	✓	✓
Non-repudiation	Repudiation			✓	✓

4.2.6 System Threats Identification

The Data Flow Diagram (DFD) in Figure 4.1 shows the graphical representation of the AV systems with the relationships between elements, such as the data flow (which is the communication data), data stores (which contain the in-vehicle and out-vehicle

storage data and databases), process (which are the functions of various components), external entities (which implies data usage, access by external services providers and stakeholders). The DFD indicates the trust boundaries with a border between trustworthy and untrustworthy elements. The oval shape elements show the functional components, and the rectangular elements show the external entities with links to data storage and data flow. Table 4.1 gives the overview of the data flow diagram (DFD elements) with the security threats against the system's protection. DFD elements are marked with ✓to map it to their threat and follow a threat tree pattern to choose the threat applicable to the specific system. Lastly, the threat tree pattern in Section 4.3.3 explains a valid attack path through its structure while presenting the potential of the threat to the system.

4.2.7 Risk Definition

Using threat modelling, the risk level can be determined, control of the significant security threats can be prioritised, and map each threat to system vulnerabilities.

4.2.8 Control Implementation

This implementation proposes the appropriate countermeasures and defences to mitigate the risk level presented by the top threats. This Section aims not to suggest security controls but to use the outcome of this analysis to justify the security threat assumptions leading to the privacy threat modelling.

4.3 AV Privacy Threat Modeling

This modelling maps privacy threat types to the DFD elements using the LINDDUN methodology [50]. It identifies relevant privacy threats to the system using a privacy tree pattern to illustrate the AV system. The methodology also provides a robust privacy conceptualisation to target system-specific modelling and compliance with standard privacy terminologies. Figure 4.2 presents the privacy threat framework and steps taken from identifying threat types, mapping the DFD elements to the threat types, identifying vulnerable conditions, assessing risks, analysing mitigation strategies and selecting privacy enhancing techniques for the system.

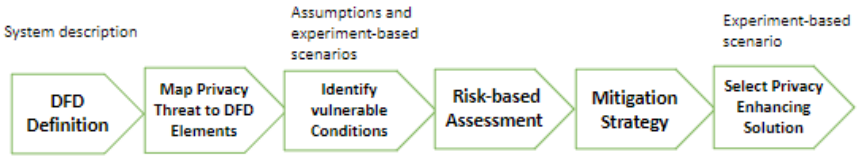


Figure 4.2: Privacy threat framework [50]

4.3.1 DFD Definition

The DFD defines the flow of data through the AV system. All the elements in the DFD are represented with a shape. Figure 4.1 already states the required DFD of the system. Other steps of the threat modeling will refer to the DFD in Figure 4.1.

Table 4.3: Privacy threat properties, types, and threat tree notation.

Privacy Property	**Privacy Threat Type**	**Threat Tree Notation**
1. Unlinkability: refers to concealing the link between two or more Items of Interest (IOIs, e.g., subjects, information, actions)	Linkability: refers to revealing the links between two or more IOI. For example, linking objects in images such as popular buildings and vehicle plate numbers to a vehicle user or a location.	The linkability of a DFD element refers to a pair (a_1, a_2), where $a \in E, DF, DS, P$ is the linkable IOI. For instance, an attacker can relate a set of DS elements, e.g., images, to an entity E, e.g., user's location, formulated as this pair (DS, E).
2. Anonymity and Pseudonymity refers to hiding a subject's identity associated with an IOI by using random variables called Pseudonyms instead of their real name. For example: generating a random name for vehicular authentication and identification in V2V.	Identifiability: refers to explicitly identifying a subject associated with an IOI. For example, Identifying a vehicle by its beacon messages in V2V.	The identifiability of a DFD element refers to a pair (a,b) where the identifiable subject $a \in E$, and the attribute identifiability relates to $b \in E, DS, DF, P$ For example, identifying an entity within a set of entities (E, E) or identifying a vehicle from its messages (E, DF).
3. Plausible Deniability: refers to not being able to prove that a subject has done something. For example, An attacker cannot prove that a set of camera data belongs to a vehicle or that an encrypted communication message belongs to a vehicle.	Non-repudiation: refers to proving that a subject has done an action. For example, an attacker can prove that a vehicle travelled past a location through communication messages (security signatures) or other sensor data.	Similar to linkability and identifiability, non-repudiation at each DFD element refers to a pair (a,b) where $a \in E$ is the non-repudiating subject and $b \in DS, Df, P$ is the attribute it relates to.
4. Undetectability and Unobservability refer to hiding a user's activities. For example, an attacker cannot correlate images or videos with a user's location, travel patterns, or trajectories.	Detectability: refers to revealing the subject's IOI For example, correlate an event with a subject. Infer location details from trajectories or camera data.	
5. Confidentiality: refers to protecting IOI by preventing unauthorised disclosure.	Information disclosure: refers to revealing sensitive information that gives enough information about an entity to an unauthorised person.	
6. Content Awareness: refers to the unawareness of users about the type of data collected, stored, and processed by the system. There is no specific rule guiding consent to process data collected by the AV. Compliance with privacy ensures good levels of trust and engagement between the users and the system.	Content unawareness: refers to the unawareness of the data the system collects.	

4.3.2 Privacy Threat Mapping to DFD Elements

This Section presents the components of privacy threats described by the methodology against privacy properties. Then map the threat categories to the DFD elements. Table 4.3 shows seven types of threats and the privacy properties they exploit. The

table explains the privacy threat types and analyses the threat tree notations to identify relationships between the threats and the elements. For example, linkability refers to revealing the links between two or more Items of Interest, IOI (e.g., subjects, information and actions). So, the possibility to link the DFD elements of a pair (a_1, a_2), where a $\in$ E, DF, DS, P is such that either two or more elements may be related. In terms of the relationships, an attacker can relate a set of DS elements, e.g., images, LIDAR data, GPS data, to an entity E, e.g., user's location, formulated as a pair (DS, E). Therefore, related linkability combinations could be written as a pair.

For identifiability threat, which refers to explicitly identifying a subject with an IOI, the identifiability of a DFD element referring to a pair (a, b) where the identifiable subject a $\in$ E and attribute to associate with it relates to b $\in$ E, DS, DF, P. Therefore, we can identify an entity within a set of entities (E, E), e.g., identifying a vehicle's subject or identity from a service provider's communication and authentication messages. Also, identifying a vehicle from its message can be written as a pair (E, DF).

For non-repudiation threat, which refers to being able to prove that a subject has done an action, similar to linkability and identifiability, at each DFD element refers to a pair (a, b) where a $\in$ E is the non-repudiating subject and b $\in$ DS, DF, P is the attribute it relates to. For instance, an attacker proves that a vehicle travelled past a location through its communication messages, camera data, or other sensor data can be written as either (E, DS) or (E, DF).

For detectability and information disclosure threats, which generally refer to revealing

a subject's IOI, each DFD element can be written as a pair (a, b), where a is the subject and b is any sensitive information relating to a subject.

The DFD element marked ✓indicates a potential privacy threat to the system in Table 4.5. For example, entities, data flow, and data stores are susceptible to linkability threats. Four elements shown in the table are susceptible to detectability and policy noncompliance threats.

Table 4.5: Threat types and the DFD elements they attack

Privacy Property	**Privacy Threat Category**	**E**	**DF**	**DS**	**P**
Unlinkability	Linkability	✓	✓	✓	
Anonymity and Pseudonymity	Identifiability	✓	✓		
Plausible Deniability	Non-repudiation			✓	
Undetectability and Unobservability	Detectability	✓	✓	✓	✓
Confidentiality	Information Disclosure		✓	✓	
Content Awareness	Content Unawareness				
Policy and Content Compliance	Policy/Content noncompliance	✓	✓	✓	✓

4.3.3 Identify Vulnerable Conditions via Threat Tree Patterns

This Section gives details of the privacy threats to the AV system and the vulnerability preconditions for each category of the threats. Figure 4.3 shows the AV system privacy threat, where the circle shape shows the root threat; the rectangle shape shows the substantial threat, and the arrows indicate an 'AND' / 'OR' relationship between the threats. The analysis in this Section assumes that the DFD elements in the trust

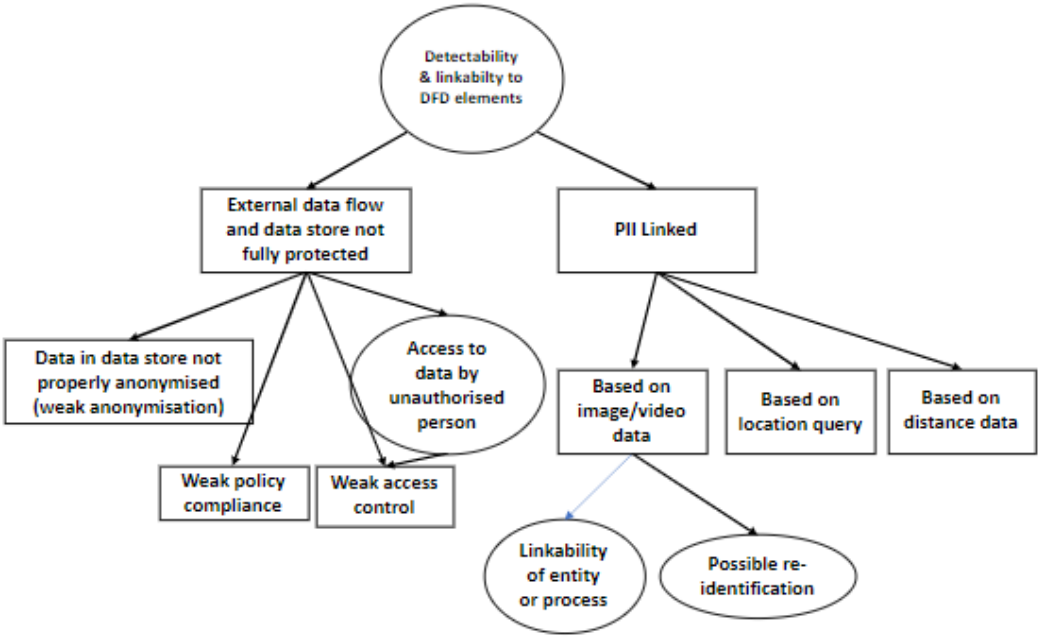


Figure 4.3: AV System Privacy threat tree pattern

boundary (as depicted in Figure 4.1) are trustworthy, i.e., a data privacy breach is less likely to happen because of its strong security assumptions and network isolation from external influence. Therefore, the analysis focuses on the elements outside the trust boundary in building the privacy threat tree. The threat tree focuses on linkability and detectability threats to the DFD elements. These are highlighted as the most significant threats that motivate an attack on the system. They mainly target the vulnerabilities in data stores and flows of the system. The first vulnerable precondition of vulnerability applies to a data store or flow not fully protected—for example, weak encryption and pseudonyms in system applications, also weak access control implementation of PTTs for communication with external devices and systems.

Previous works show various attacks to link encrypted packets and pseudonyms from sniffed communication data in a V2X setting [57]. These attacks may lead to infor-

mation disclosure threats, e.g., vehicle location beacon disclosure. Also, the system's weakness could lead to unauthorised access to data in the data store or flow. For example, an attacker may gain access to an AV cloud data storage to access camera data and other sensor data.

The second vulnerability precondition applies to weak data anonymisation in the data store, for example, camera data exposing sensitive information such as pedestrian faces in images, vehicle plate numbers, popular buildings, signposts, etc. This sensitive information could breach users' location privacy by using a geo-localisation tool for an attack. Section 4.4 shows an experimental analysis for inferring location details from camera data. The Section justifies the implication of weak data anonymisation and the need for efficient anonymisation. Lastly, the third vulnerability precondition applies to poor policy compliance. An example of this weakness is when authorised service providers or data recipient process data for the wrong reasons that could breach the user's privacy. These weaknesses could also lead to unauthorised data access due to poor access control and sharing.

Lastly, the third condition of weakness applies to poor policy compliance. This condition can happen when authorised service providers process data for the wrong reasons that could breach the user's privacy.

4.3.4 Risk-based Prioritisation

This analysis focuses on the most significant problems, potential privacy impacts, and technical measures to address the problem. This Section does not dive deep into risk-based assessment but maps the identified risks to the privacy impacts. We generally add the likelihood of an attack scenario with the impact to calculate the risks. The higher the risk, the more significant the privacy impact. Privacy impact implies the harms on the subject caused by privacy breaches. For instance, if an attacker detects the travel pattern of a vehicle through its camera data, this could cause privacy harm, such as theft or home invasion.

4.3.5 Mitigation Strategy and PET Selection

This Section explains the privacy requirements for the identified threats in the analysis and chooses mitigation strategies and privacy-enhancing techniques based on the privacy objectives. The most significant threats to elements outside the trust boundaries are detectability, observability and linkability. Then the conditional vulnerabilities of the system are weak data protection, weak access control and weak anonymisation. The privacy objectives now are to protect the data stores and flows adequately. Then select a more robust access control for data communication with service providers and stakeholders. Lastly, ensure a more vigorous anonymisation technique to conceal private information in camera data for storage and analysis use cases. *Hajny et al.* proposed several PETs for technical measures covering different aspects such as data anonymisation, privacy-preserving data, communication protection and authen-

tication [155]. Focusing on anonymisation techniques suitable for AV privacy/utility requirements, *Xiong et al.* used an efficient privacy-preserving method for AV camera data. The analysis in section 4.4 shows that vehicle locations could be geo-located through inference by 20% more than other attacking tools. A location privacy breach will likely happen if *Xiong et al.* AV camera data storage technique is used. Therefore, this thesis proposed a better privacy-preserving generative technique for AV camera data for storage and analytics in Chapter 5. The proposed technique ensures that a robust geo-localisation method reduces the detection of sensitive objects in the camera data, and non-sensitive objects retain their visual quality for analytical purposes.

4.4 Location Inference Attack

The estimate of image geolocation has become more accurate due to the formulation of robust computer vision techniques for object recognition. Publicly accessible street-view images have also made geo-localisation tasks possible [128]. AV vendors contribute significantly among major online street-view image contributors [3]. There is a motivation to study location inference attacks on AV camera data, relying on geo-localisation approaches to match a specific query image data to a corresponding reference data in a database. Although existing city-wide geo-localisation techniques show tremendous performance on street-view data, their accuracy significantly drops when tested on distorted query data (i.e., GAN-based privacy-protected data). The technique's performance is affected due to the reduction of object recognition in the query image, which results in mismatching. For instance, *Zamir et al.* could localise

an original query image (I.e., street-view image without privacy protection) by a variable accuracy of over 70% but produced a reduced performance of 20% and between 4% and 5% on ADGAN 1 (Auto-driving generative adversarial network) and ADGAN II generated data, respectively [13]. *Zemene et al.* could also localise an original query image better than [13], offering about 20% improved accuracy, yet an accuracy of between 7% to 10% for the distorted query data [11]. While these techniques and other [12, 134, 156] city-wide geo-localisation techniques produce satisfactory results for localising original query images, their results on distorted images are very low. This result means that it is not likely to get a matching image or estimate a location close to the actual location of the ground truth image.

The formulation of privacy-preserving auto-driving data, specifically using GAN-based methods, reduced geo-localisation chances or made localisation almost impossible [14]. In the original image data, the intuitive steps taken to enforce privacy preservation involve visually reducing sensitive objects such as buildings, road signs, road information, signpost, surrounding vegetation, road vehicles, etc. Since a geo-localisation approach relies on recognising objects and detecting stable features for image matching [134], privacy-preserved techniques (distortion techniques) reduce its efficiency. Therefore, this thesis improves the geo-localisation approach of *Zemene et al.* [11] to attack the location privacy of distorted AV-generated data by detecting and matching stable features. It further analyses attacking steps to infer travel location patterns of the image dataset assumed to have been accessed from an AV.

The technique [11] defines a dynamic nearest neighbour selection to improve the near-

est neighbour selection of multi-KNN. It proposes a dominant clustering approach to match compact sets of stable features. The geo-localisation approach in this thesis follows a similar implementation with the following difference: It defines a distinctive nearest neighbour selection and formulates an optimisation problem for pairwise clustering to handle the complexity of the large graph and improve feature matching.

LIFT technique takes a query image as an input, extracts the local object features using SIFT [12], and distinctively retrieves each query feature's nearest neighbour (NN). Pairwise clustering (PC) is adopted as the primary technique for feature matching and a robust similarity metric for choosing the best matching reference image based on their global features. The PC approach presents image-matching as a clustering problem to learn the discriminative features between the original and distorted images (GAN-generated image data). This technique suits this problem because it can organise large NNs in a graph. It can also handle distinctive selections of query features with consistently stable NN, resulting in robust feature matching. The technique also connects to game theory, allowing the efficient use of game dynamics, such as replicator and infection-immunisation dynamics [157]. The approaches proved to have a linear time/space complexity, making the technique fast. For this reason, we can run the technique several times for the image-matching task to yield better accuracy.

Google street view data are good sources of reference data for geo-localisation analysis [127], and they provide street-view images of spherical 360-degree panoramic views approximately 12 meters apart. This analysis uses a sample of each location of several European city areas. These samples are similarly distributional to the collected

test query data. In recent studies, a drawback that affects accuracy when using Google Street View for geo-localisation is the low-quality visuals of the distorted online images. However, the analysis experimented with other quality street scene data, such as the Berkeley dataset, to validate the technique.

The main contributions of this Chapter are as follows:

1. Design LIFT, a robust geo-localisation approach, pairwise clustering based, for localising distorted query images (i.e., images generated from a privacy-preserving model, e.g., Auto-Driving GAN) from a reference database of large-scale geo-tagged images. Existing studies have only used original or Gaussian blur data for their analysis.

2. Formulate a distinctive query feature selection method that arranges features according to their informative values. This procedure provides enough informative features and a large graph for the next stage's primary and secondary feature matching.

3. Formulate a robust quadratic function with pairwise clustering for computing the image's local and global features for efficient feature matching.

4. Train the reference database using original images from Google Street View and Berkeley open-source data. To generate the blurred sample base images that form the image pyramid.

5. Analyse the threat matrix to capture attack potential, motivation, and impact.

4.5 Geo-localisation Approach for Noisy Images

This describes the proposed geo-localisation approach in Figure 4.4. The reference feature database contains SIFT local features [12] extracted from images taken from Google Street view. We organise the original reference image features and noisy image references in a k-means tree data structure. The original reference features refer to the features of the reference image without noise. In contrast, noisy reference features refer to features of the reference images with added noise (i.e., masked noise in ADGAN paper). Privacy-preserved images from an ADGAN model [14] are used for query images. Then extract distinctive SIFT local features, referred to as query features. Select the distinctive nearest neighbour from their corresponding query feature for each. Stop the selection of NN for a given query feature when the similarity ratio between the NN and the next is lower than a threshold (discussed in Section 4.4.2). By this, the last NN is discarded because it is less similar to the previous NN. Selected query features are arranged in the database depending on how informative they are; less informative features appear at the rear of the set. Next is to describe the problem of finding matching reference features to query features by predicting the original image features using pairwise clustering to extract the most compact set. In the final step to finding the best matching reference image, a voting scheme is used to choose the best matching image using the combination of local and global features in an affinity propagation approach.

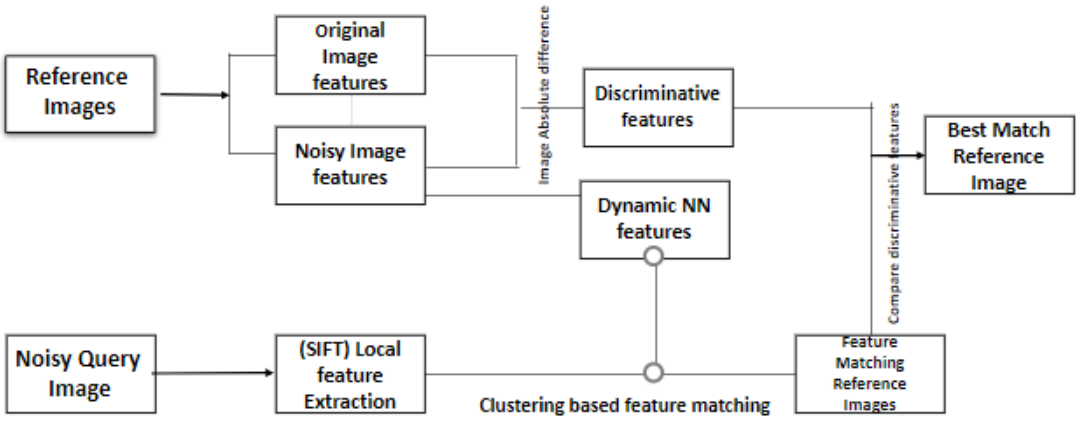


Figure 4.4: Geo-localisation approach for noisy image queries that matches large distinct SIFT features using a pairwise clustering-based approach and the best reference image by voting

The proposed attack model, LIFT, comprises five steps for geo-localisation.

- Train the reference database with original images and generate a distorted base image using an ADGAN model.
- Compute discriminative features of the original and base images by finding their absolute difference.
- Use the base image (SIFT) features for image matching and organise the extracted features in a k means tree data structure.
- Use test query image for matching by extracting SIFT features and matching features with the reference base images to produce the matching reference images.
- Use pairwise clustering to compute the weights of the local and global features and extract a compact set.

- The best matching image is further processed by computing the similarities between the discriminative features of the query and reference. The one with the closest value will be chosen as the best match.

4.5.1 SIFT Composition

SIFT technique follows several stages: scale-space extrema detection, keypoint localisation, orientation assignment and keypoint descriptor to compute and generate the image features. To form the scale space for building an image pyramid, first generate a base image that is doubled in size and blurred using the L_1 distance location privacy in ADGAN. Then compute the octaves in the image pyramid. A function generates the blurred image and frequently downsamples the base image. Subtract adjacent pairs of the images using the absolute different function to generate a pyramid of discriminative GAN images. Keypoints are identified and extracted from the base image. Clean up keypoints duplicates and convert the keypoints into the original image size. These keypoints transform into a descriptor that allows their comparison with an image query. Now we show how to sample the scale space without adding gaussian blur and incrementally convolving the blurred image query with Gaussians to produce images separated by a constant k. Let the scale space be $L(x, y, \sigma)$, where x and y are the image width and length, respectively, and σ is the signal noise set to zero to avoid adding Gaussian blur. We produce L from a variable scale Gaussian, $G(x, y, \sigma)$, where σ is set to zero, with the input $I(x, y)$:

$$L(x,y,\sigma) = G(x,y,\sigma) * I(x,y) \tag{4.1}$$

Where $*$ is the convolution operation in x and y and

$$G(x,y,\sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}} \tag{4.2}$$

For efficient and stable keypoint location detection in the scale space, this thesis uses scale space extrema in the difference of Gaussian function convolved with the image, $A(x,y,\sigma)$, which can be computed from the difference of two nearby scales separated by a constant multiplicative factor k:

$$A(x,y,\sigma) = \|(G(x,y,k\sigma) - G(x,y,\sigma))\| * I(x,y) = L(x,y,k\sigma) - L(x,y,\sigma) \tag{4.3}$$

To compute the number of octaves, half each octave of scale space into an integer number s of intervals so that $k = 2^{\frac{1}{s}}$. Generate $s+3$ images for each layer in the stack of blurred images for each octave and produce another $s+2$ for one blur step before the first image in the layer and another blur step after the last image in the layer. This step will help create the difference in the Gaussian image pyramid in the final step. Many of the same blur values are produced by comparing the two neighbouring layers to cover all blur steps.

4.5.2 Distinctive Nearest Neighbour Selection

This procedure explains how the query features' nearest neighbours (NNs) are selected and the database arrangement to include the most informative features at the top and the less informative features at the rear. Features such as foliage and ground plane convey hidden contextual information for secondary matching if the distinctive features do not provide enough matching performance. For ease of reference, Table 4.7 summarises the frequently used notations in this Chapter.

Table 4.7: Frequently used notations 1

Notation	**Description**
N	Number of query features
q^i	ith query feature
G	Graph
V	Finite set of nodes
$v_{m,i}$	mth nearest neighbour
E	Set of edge
u	Node weights
w	Edge weight
A	Object features similarity matrix
γ	Scale value
$\varphi(.)$	Operator returning the global descriptor
ξ	Operator returning the local descriptor
θ	Minimum threshold from similarity metrics

Let N be the number of query features detected in a query image with corresponding

nearest neighbours NN. For each ith query feature q^i, let $v_{m,i}$ be the mth nearest neighbour, where $m \in \mathbb{N} : 1 \leq m \leq |NN^i|$ and $i \in \mathbb{N} : 1 \leq i \leq N$. NN^i represents the set of NNs of the ith query feature and $|.|$ is the set cardinality.

Add to the nearest neighbour set, $v_{m+1,i}$ of the ith query feature if the NNs are similar. The similarity of two consecutive NNs is measured and add $(N+1)$th NN if the NNs are greater than θ, then stop adding NNs when a less similar object feature is detected (i.e. below θ).

The selection can be formulated as follow:

$$\mathbb{V}^i = \begin{cases} \mathbb{V}^n \cup v_{m=1,i} & \text{if} \frac{||\xi(q^i)-\xi(v_{m,i})||}{||\xi(q^i)-\xi(v_{m+1,i})||} > \theta, \quad m+1; \\ stop & otherwise \end{cases} \tag{4.4}$$

With an input of ith query feature and its detected NNs, $q^i = v_{1,i}, v_{2,i}, \ldots . v_{|NNi|,i}$, let's distinctly generate an output $\mathbb{V}^i = v_{m,i}$, where m is initialized to 1. The selection procedure keeps adding NNs if $v_{m,i}$ and $v_{m+1,i}$ are not very discriminative but stop when a discriminative neighbour is detected based on the θ value. This procedure is repeated for all query features and returns a set of distinctly selected nearest neighbours.

The next procedure identifies the features based on how informative they are and arranges them according to this formation. Then coarsely detect features such as moving objects (e.g., pedestrians, vehicles) and append them to the end of the set. This allows for keeping enough information in the tree for the feature-matching stage. The most

informative features at the top of the tree are selected for feature matching. However, in the case of low feature-matching performance, the less informative features at the rear provide additional context.

The selection is formulated as follows:

$$Q = \begin{cases} Q = q^1, ..., q^N & \text{if} \frac{||\xi(q^i) - \xi(v_1)||}{||\xi(q^i) - \xi(v_{|NN^i|,i})||} > \beta; \\ q^{i+1} & otherwise \end{cases} \tag{4.5}$$

Where ξ represents an operator that returns the argument node's local descriptor, the query features are arranged to depend on their threshold value θ, such that the most informative features are placed at the top. The less informative features are appended to the rear of the set. By this formulation, we choose the top few query features for feature matching or use the whole query feature set for matching in a less-performing step. Lastly, set θ to 0.5, a minimum threshold derived from comparing similarity metrics of the distorted with the corresponding original image. When θ is set to 0.7 according to [11], many features did not make the selection process. This is because most distorted data features are less recognisable using that threshold. Setting θ to 0.5 allows the retrieval of more features for the next step, which means that the features are 50 percent similar to the reference image feature. Also, set β to 0.9 to retain enough query features for the next stage. This value is set to compare the resulting selection values and arrange them in the graph according to their informative context. Less informative features are not pruned as described in [11] but are appended to the rear

of the graph.

4.5.3 Distinctive Feature Matching Using Pairwise Clustering

The clustering problem extracts a compact group from a large set of objects using pairwise similarities. It can be formulated as an edge-weighted generalisation of a clique to extract a coherent set of image features. The clustering approach can be formally described as a graph $G = (V, E, u, w)$, where $V = 1, \ldots, n$ is a finite set of nodes, a set of edge $E \subseteq V * V$, node weight $u : V \longrightarrow \mathbb{R}$, and an edge weight $w : E \longrightarrow \mathbb{R}$.

The graph node V is the image feature to be clustered with edges representing the neighbourhood relationship between the features and weights accounting for the similarity among connected features.

Let A denote object features similarity matrix, such that $A_{ij} = w(i, j)$ for all $i, j \in V$. The notion of a coherent set follows that a non-empty subset of objects $C \subseteq V$, such that $i \in C$ and $j \in C$ where i and j are nodes in the graph G, we have:

$$\Phi_c(i, j) = A(i, j) - \frac{1}{|c|} \sum_{i \in c} A(i, j) \tag{4.6}$$

Where $|c|$ denotes the cardinality of C. Also, note that $\Phi_c(i, j)$ can be either positive or negative. Then, assign each node $i \in C$ to a weight defined recursively as follows:

$$W_c(i) = \begin{cases} 1 \quad \text{if} |C| = 1, \\ \sum_{j \in c \setminus i} \Phi_{c \setminus i}(i,j) W_{c \setminus i}(j) \quad otherwise \end{cases} \tag{4.7}$$

Where $c \setminus i$ is set S without the element i, which means that $W_c(i)$ measures the overall similarity between the node i and the node of $c \setminus i$. Thus, a positive $W_c(i)$ means that adding i into its neighbours in C will increase the internal coherence of the set. In contrast, the overall coherence decreases with a negative value. The total weight of C is denoted as

$W(c) = \sum_{i \in c} W_c(i)$

So, a non-empty subset of objects $C \subseteq V$ such that $W(T) > 0$ for any non-empty $T \subseteq C$ is said to have a compact set if:

$W_c(i) > 0$ for all $i \in C$, $W_{c \cup i}(i) < 0$, for all $i \in C$

The above conditions agree with the two main properties of a cluster: internal homogeneity (which has an element belonging to the cluster with high mutual similarities) and maximality (which is a cluster that cannot be further extended by introducing an external element).

Let's formulate a relation between pairwise clustering (PC) and strict local maximisers to derive a standard quadratic optimisation problem.

$$\max_{e\in[0,1]^n} f(x) = \frac{1}{2}x^T Ax \tag{4.8}$$

Where A is a square matrix of order n with the following formulation.

$$A(i,j) = \begin{cases} 0 & i = j \\ w_{\{}ij\} & \{i,j\} \in E \\ -\hat{w}_{\{}ij\} & \{i,j\} \notin E \end{cases} \tag{4.9}$$

Note that x is a vector of features corresponding to vertices of G and $\hat{w}_{\{}ij\}$ is defined as

$$\hat{w}_{\{}ij\} = \max\left\{ \sum_{k\in N(i)} w_{ik}, \quad \sum_{k\in N(j)} w_{jk} \right. \quad + \xi \forall i,j \notin E, i \neq j \tag{4.10}$$

For an arbitrary small $\xi > 0$. Any global optimal solution of the quadratic optimisation problem is the characteristics vector of the compact set of the graph G. Note that $N(i)$ is a neighbouring vertex of i in G defined as $N(i) = j \in V | \{i,j\} \in E$. Also, note that A is symmetric and always indefinite by the definition of an undirected graph.

4.5.4 Feature Matching Similarity Function

The similarity function learns a metric that matches the query feature to the reference features. It uses the global and local features of the images to select correct NNs from

the large graph G, which forms a highly compact set.

The set node V selected from the distinct NN feature selection phase represents all NNs for each query feature. Edge set $E = (v_{m,i}, v_{n,j})|i \neq j$ represents all connected nodes in G, if their corresponding query features are not the same. The edge weight reflecting similarity between the linked vertices adopts a similarity function from [158], consisting of a Gaussian kernel similarity measure formulated as

$$w(i,j) = exp(-\frac{||\varphi(i)-\varphi(j)||_2^2||}{2\gamma^2}) \tag{4.11}$$

Where γ is a scale value set to 2^7 and $\varphi(.)$ is an operator which returns the global descriptor of the parent image of the argument node. The similarity function between the edge weights $v_{m,i}$, $v_{n,j}$ can be represented as

$$w(v_{m,i}, v_{n,j}) = exp(-\frac{||\varphi(v_{m,i})-\varphi(v_{n,j})||_2^2||}{2\gamma^2}) \tag{4.12}$$

and also represent the similarity between nodes $v_{m,i}$ and $v_{n,j}$ as

$$v(v_{m,i}) = exp(-\frac{||\xi(q^i)-\xi(v_{m,i})||_2^2||}{2\sigma^2}) \tag{4.13}$$

to show how similar the node $v_{m,i}$ is with its corresponding query feature regarding its local features. The global feature in this expression uses CNN in [13], because it performed better than HSV histogram [159] and GIST [160].

Next, compute the optimisation problem by the following formulation to represent the similarity between the query feature and the corresponding reference image.

$$\max_{e\in[0,1]^n} f(x) = \frac{1}{2}x^T Ax + b^T x \tag{4.14}$$

Where A is an affinity matrix representing the global similarity between reference images and b is the node score. The formulation in (4) can be substituted for

$$A(v_{m,i}, v_{n,j}) = \begin{cases} 0 & i = j \\ w\{v_{m,i}, v_{n,j}\} & \{i,j\} \in E \\ -\hat{w}\{v_{m,i}, v_{n,j}\} & \{i,j\} \notin E \end{cases} \tag{4.15}$$

And $b(v_{m,i}) = \xi(v_{m,i})$

4.6 Experimental Results

To effectively test the performance of LIFT, datasets are collected for evaluation, and state-of-the-art techniques in geo-localisation are compared with the result.

Dataset for Evaluation

For LIFT evaluation, this study uses the reference dataset, 200k publicly available Google Street View images [6] and another 200k street view images from Berkeley's open-source AV repository [91]. The Google Street View dataset covers a 360-degree

panoramic view of a certain location in four separate images. In contrast, the Berkeley dataset shows a better scene view in one image. The Berkeley images are sequences of frames generated from the video captured in an AV driving experiment in the United States, including Brooklyn and Boston states. This dataset provides high scene quality and a canonical view.

Figure 4.5: Training and test data extract from the repositories [6, 91]

For the query data to match the reference data, 500 geo-tagged images are collected from similar locations of different repositories such as Flicker, Pisca and Panorama. Each image is manually checked to match several locations in the reference database. The original downloaded images are transformed into privacy-preserving images as the base images for computing keypoints and descriptors. Figure 4.5 shows street view images of different locations from the training and test dataset.

Comparison of the Geo-localisation Result

This analysis shows the performance of the proposed approach on the noisy dataset and presents a quantitative comparison of the state-of-the-art techniques. Figure 4.10 shows a comparison of the approach with the baseline methods, error threshold on the x-axis in meters and the performance of the test set localisation performance on

the y-axis. Figures 4.6, 4.7, 4.8, and 4.9 depict the feature matching performance of the methods. The red lines match the detected keypoints to their similar reference keypoints. Table 4.9 presents the performance of the geo-localisation techniques in percentages within a specific error threshold.

In Figure 4.6, the original image on the left side matches the features to the exact reference features on the right. Many features in the image are detected and correctly matched. This shows that LIFT distinctly matches the original image with good performance. In Figure 4.7, the feature-matching efficiency of GMCP on distorted query data is shown. The method detected many features, which are false positive results. However, it did not localise the exact reference image. Likewise, in Figure 4.8, DSC detected some stable features in the distorted image but could not localise the exact reference image. However, DSC showed better feature-matching accuracy than CNN and GMCP because the exact image was among the NNs in its selection process. In Figure 4.9, LIFT detected some stable features and matched them to many reference images. However, LIFT localised the exact matching reference image with its stable global and local features, as shown in the image. This indicates that LIFT performed better on the specific distorted data than the others.

Table 4.9: Geolocalisation performance of the techniques

Geo-localisation Technique	**Localisation Performance on Distorted data within**	
	60 meters	**300 meters**
CNN-based [17]	4%	4.5%
GMCP-based [13]	6%	7%
DSC-based [11]	10%	12%
LIFT	20%	29.5%

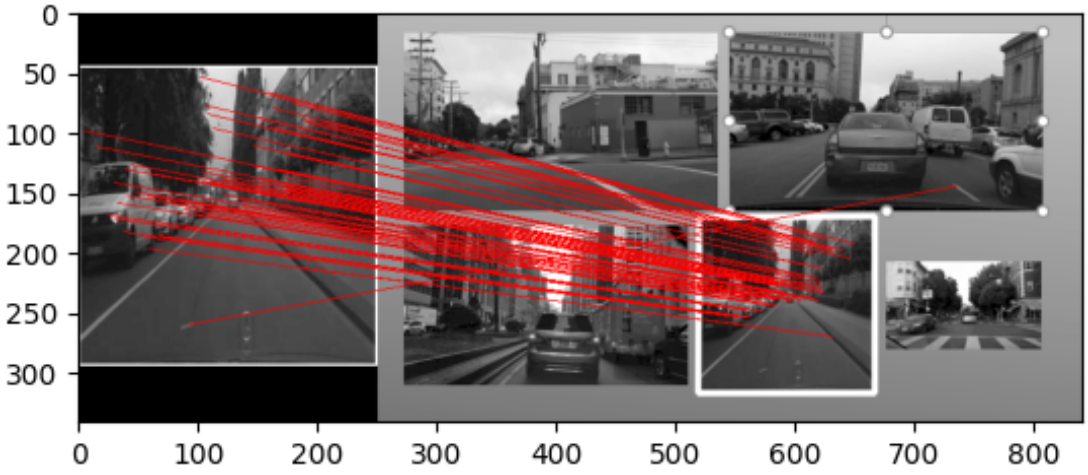


Figure 4.6: Feature matching of original images without privacy protection.

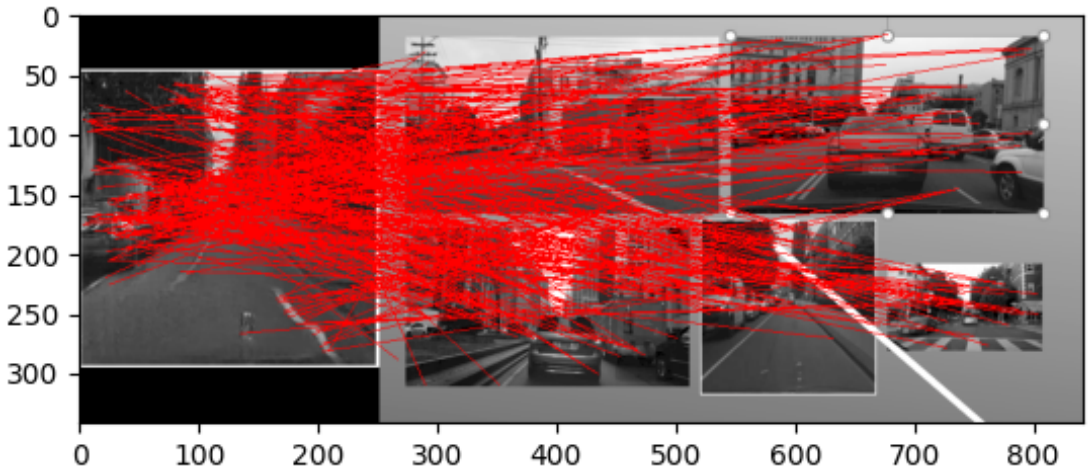


Figure 4.7: GMCP-based feature matching using distorted data.

The following describes the baseline methods used for the comparison and evaluates their performance.

- CNN-based geo-localisation: trains the dataset using a neural network in a supervised setting to match features with its query image. The black line in Figure 4.10 depicts the localisation results on the dataset. The method learns the less informative features from the query set in matching the learned feature. This performance is due to the neural network's inability to learn enough local features from the distorted query set. That is, object recognition performance was low. It localises with lower percentages and errors. The accuracy of geo-locating the distorted query set with the method is about 4% within 60m and 4.5% within 300m. Hence the method did not perform well in localising the images.

- GMCP-based geo-localisation: solves feature matching problem using generalised maximum clique problem and a voting scheme to select the best matching reference image. The approach localises much better than the CNN method,

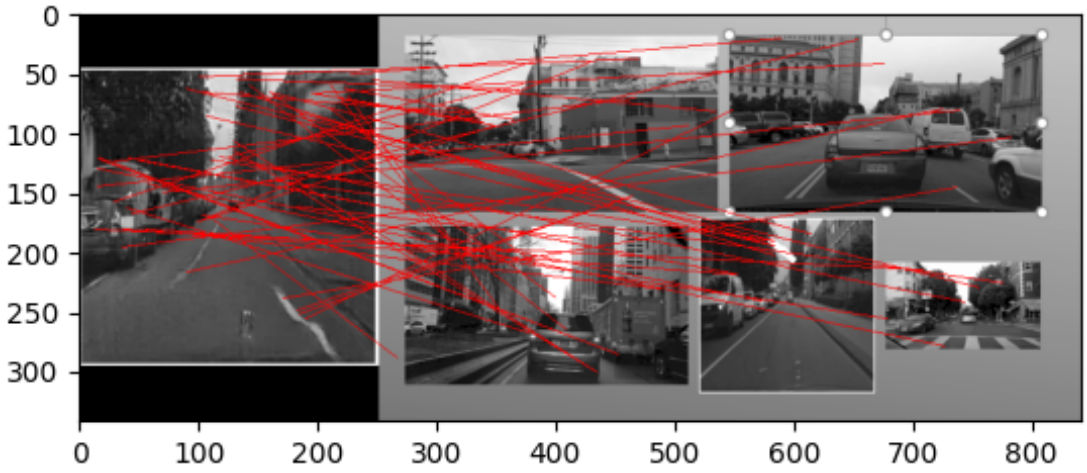


Figure 4.8: DSC-based feature matching using distorted data.

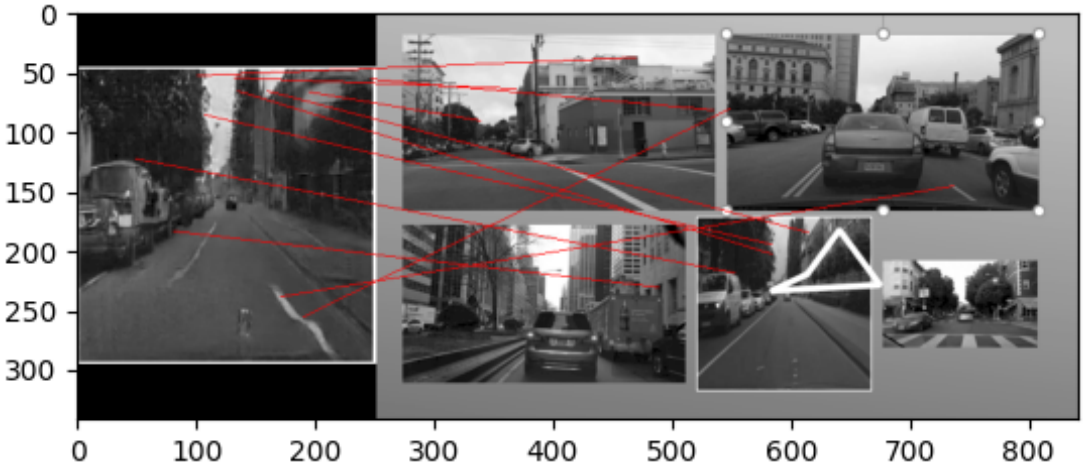


Figure 4.9: LIFT-based feature matching using distorted data.

with an accuracy of about 6% within 60m and 7% within 300m. However, its performance could not be improved due to their formulation of a fixed nearest neighbour selection. Their robust distance measurement enhanced the performance and made it perform better than the CNN approach.

- DCS geo-localisation: solves feature matching problem with an improved localisation as nearest neighbour increases using dominant set clustering and con-

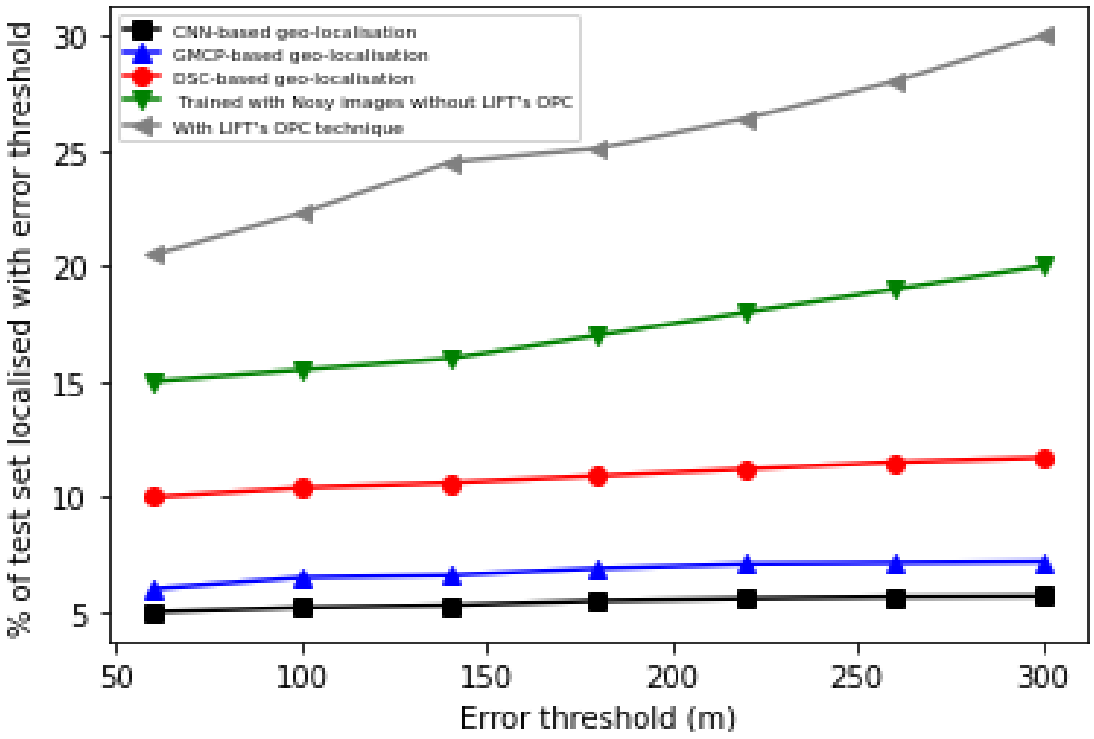


Figure 4.10: Geo-localisation performance comparison.

strained DSC-based post-processing. It localises with an accuracy of about 10% within 60m and 12% within 300m. The improved accuracy results from the robustness of clustering on larger feature objects of the inliers and outliers of the images.

- LIFT-based geo-localisation (the approach in this thesis): geolocalises noisy images by training on the GAN-based generated images to match their distinct features to a similar query image. The image's discriminative features improve feature matching using pairwise clustering to organise and train on a large graph. The method outperforms the baseline method, as shown in Figure 4.10. 10% improvement is achieved by just training with the noisy reference images. Another 10% localisation accuracy is achieved within 300m using Google Street View

images after applying the LIFT-based technique to handle the large features. More accuracy is achieved using the Berkeley dataset. The accuracy could be improved if there were more viewpoints of a certain location, such as Google Street view images. However, the improved accuracy results from the initial image quality and enough viewpoint for a certain location.

4.7 Discussion on Location Inference Threat on AV Camera Data

The attack model with the following parameters: "Who", "Where", and "What", links the location inference threat to an AV subject. This attack model shows the motivation and capabilities of an adversary to breach the privacy of AV, with links between the parameters. The first threat model parameter, "Who", refers to a subject, which may be a driver, vehicle owner, vehicle identification number and authentication information. The second parameter, "Where", refers to a set of locations, e.g., trajectory, subject events, and places of interest (home and office addresses). The third parameter, "What", refers to the adversary's motivation; that is, what the adversary intends to do with the information—the first parameter associates with the second and vice-versa in the modelling. At the same time, the first and second parameters motivate the third.

The attacking approach in section 4.5 operationalises the "Where". In that Section, a trained and optimised localisation technique was used to localise AV distorted data with an accuracy of over 20%. This analysis and result mean that the technique could match some AV-generated images to their similar reference images, consequently leading to inferring locations, predicting trajectories, journey patterns, and an AV subject's

past and future travel patterns. In this instance, the “Where”, parameter could link the “Who”, by inference and linking attacks. This step could lead to tracking and stalking of the subject, where an adversary could be motivated to generate places of interest, leading to vehicle theft and burglary. With the high capabilities of adversaries to localise AV-generated camera data using robust geo-localisation attacking tools, leading to an increased potential privacy breach, there may be constant attacks on AV data stores and flows by motivated adversaries to access camera data.

The next step of the attack focuses on linking the trajectory with a subject or an event. Depending on the use case of the vehicle, whether for private or public use, an adversary could establish a link to the potential subject. For instance, detected places of interest, such as home and office addresses, were used to link a subject. An adversary may be motivated to determine the vehicle identification number or type for a public vehicle; an adversary may be motivated to determine the vehicle identification number or type. At the same time, the chances of linking a subject may increase with known background knowledge.

The last step establishes the adversary’s motivations. With the trajectory information, the adversary could learn the journey patterns of the vehicle and establish predicted future patterns. This step could lead to tracking and stalking of the subject. An adversary could also be motivated to generate places of interest. Consequently, this attack step could lead to vehicle theft and burglary.

Chapter 5

Formal Analysis of DeepClean Generative Clustering Model

This section presents DeepClean, a privacy-preserving generative method for AV-generated camera data to address a balanced privacy/utility trade-off in the presence of a potential location privacy threat. DeepClean uses deep clustering that combines the private Gaussian mixture model and VAE techniques. This section explains the system model, the considered threat model, and the developed approach. For ease of reference, Table 5.1 summarises the frequently used notations in this Chapter.

5.1 System Model

The original camera data is passed to the model as an input, and synthetic data is generated (as shown in Figure 5.1). The first component is the labelled DP-GMM algorithm to partition the image into k clusters, learn and predict the labelled clusters, and add Gaussian noise to the learned private object clusters. The output of this component is a noisy partitioned cluster, which is then trained in the encoder $g(x,\phi)$ to produce a latent representation z. A decoder network $f(z;\theta)$ interprets z such that a synthetic sample can be drawn from model θ.

The second component, which is the proposed DP-GMM algorithm, performs three processes: (i) partitioning the distribution into k clusters, (ii) learning and predicting

Table 5.1: Frequently used notations 2

Notation	**Description**
x	Real camera data
X_k	k clusters labelled image objects
$G(\rho_i)$	Gaussian mixture component
ρ_i	Labelled image object in the i.i.d clusters
$z \sim \mathcal{N}(\mu_j, \sigma_j^2)$	Gaussain mixture parameters
$g(x, \phi)$	Encoder network with parameters
$f(z; \theta)$	Decoder network with parameters
ε, δ	Privacy parameters
β	Failure parameter
α	Estimator accuracy of variation distance
$\hat{x}$	Distorted camera data

the labelled clusters, and (iii) adding Gaussian noise to the predicted private objects. The third component decodes the aggregated clusters by drawing a latent variable z and generating a latent distribution where a synthetic sample can be drawn. The decoded model is a deep neural network $f(z; \theta)$. The deep clustering technique contains an inference model based on the encoder of a VAE, then trained on a differentially private Gaussian mixture model for a better visual representation of non-sensitive objects and privacy control of the sensitive objects. Then the generative model produces synthetic samples with similar non-private areas to the original data, while private object areas are less similar. The main objective of the models is to learn the pixel-to-pixel transformation and generation of observed samples.

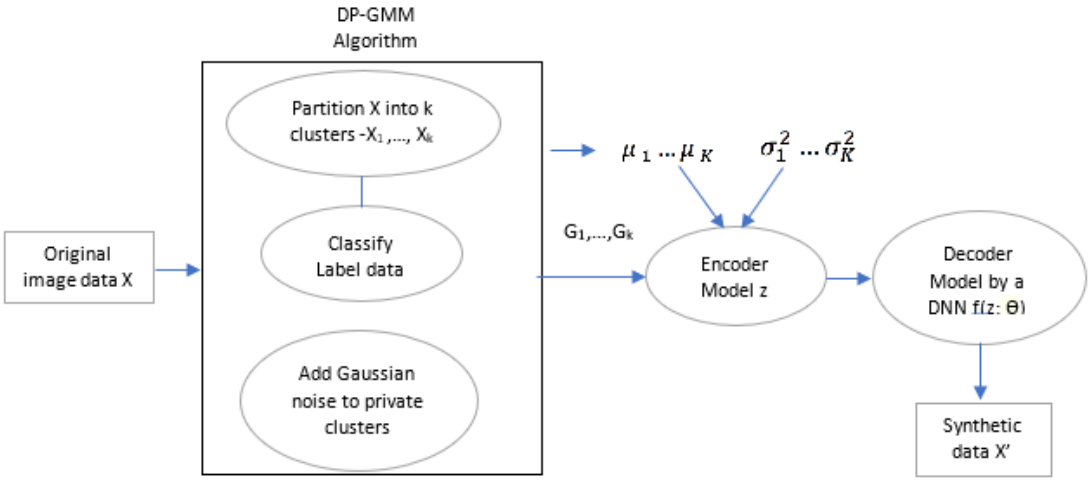


Figure 5.1: DeepClean model, showing different system components (including the combination of differentially private GMM with VAE)

Let x be a real camera image such that $x \in I$, where I is a set of raw images from real AV camera data. An image x is fed into the model M consisting of inference and generative processes, and an observable image sample $\hat{x} = M(x)$ is generated. The developed private Gaussian mixture model is applied to the sensitive clusters during inference, and the generative model produces a privacy-preserved image $\hat{x}$.

In the inference process, the private GMM partitions the labelled image objects into k clusters, $X_1, X_2, \ldots, X_k$ where each cluster is a group of similar objects in X. The GMM is trained in a supervised setting to classify the objects in the clusters. Then the GMM trains separately on each cluster; if the cluster is classified as sensitive, Gaussian noise is applied to the cluster centre, else it retains its accurate visual representation (without noise). The VAE encoder trains separately on the cluster outputs and maximises the ELBO for optimisation. In the generative process, the decoder, a deep neural network $f(z;\theta)$, decodes the embeddings to an observable, where θ is the parameter of the

resulting model.

5.2 Threat Model

The threat model considers an attacker or a curious analyst who can access a target's camera data. A vehicle user or vehicle is regarded as a target. A location inference attack can be mounted on the data with or without external multi-source information such as trajectory and distance data. The core task of the attack relies on extracting visual and contextual features, e.g., landmarks, background buildings, surrounding vegetation, and surrounding objects, from query image data. Then the features extracted from the query data are compared with the features of an already trained reference image data of a city or a group of cities (e.g., Google Street View images). If there is a match of features, the geo-localisation system returns the nearest neighbour (NN) image reference with matching features. Then a scheme is used to estimate the location of the most matching NN or even evaluate the location proximity of the multi-NN. A robust geo-localisation system must improve image feature matching and geo-location estimates.

As explained in Section 4, the developed attack tool is based on a robust geo-localisation approach. It used an optimised pairwise clustering approach for feature matching and defined a distinctive image feature selection. This geo-localisation system improves the localisation of distorted images by an accuracy of 20% compared to [12,13], which are used as the attack model in the related work.

Let's assume that if the attacker can access some AV camera data and, using this sophisticated geo-localisation system [11], she can infer vehicle location information. The attacker can learn estimated location information from the less privacy-preserved datasets generated by the state-of-the-art (e.g., ADGAN [14,15]). Figure 4.1 shows the matching reference images of a given distorted query image data (ADGAN-generated image). The exact matching image is the nearest neighbour with the most frequent occurrence (the NN with the yellow-coloured ID and frequency of 6). In contrast, the geo-localisation technique in [12] [13] cannot locate the exact match of the distorted image because of their less robust feature-matching approach.

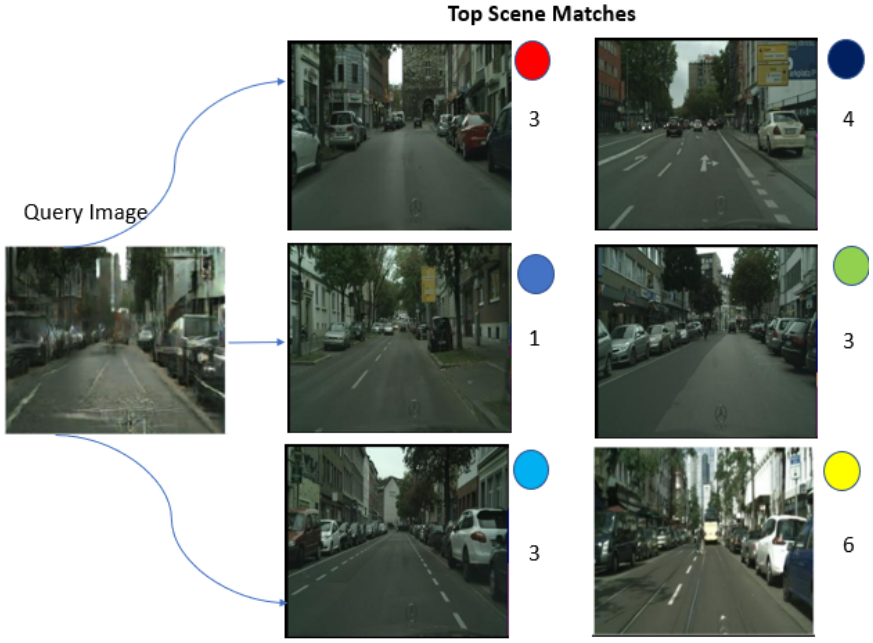


Figure 5.2: Image matching of distorted image data by the Dominant Set framework. The reference data with the yellow colour ID occur most frequently

To control the impact of this attack, the intuitive thing to do is to reduce the precision of extracting sensitive features and side-channel location information from the data.

A typical privacy-preserving approach would remove or blur sensitive objects, which is not trivial to achieve. However, data utility for analytics will be affected and not efficiently address the privacy-preserving approach for street-view images. The data generated from such a technique may be useless for data analysis, such as auto-driving navigation analysis.

This gap creates a challenge in balancing the privacy-utility trade-off. Thus, the transformed data must retain statistical structure in various non-private areas yet preserve the privacy of the private areas in the data, which is achieved through DeepClean, as explained in the following.

5.3 Description of DeepClean

As shown in Algorithm 1, a private GMM partitions X as a mixture of Gaussians with labelled clusters $G_\rho = ((G_1, \rho_1), \ldots., (G_z, \rho_z))$, where G_i can be chosen from the mixture component $G(\rho_i)$, such that ρ_i is a labelled image object in the i.i.d clusters. A cluster G_i is an output of the private Gaussian mixture partitioning algorithm on X. Then compute an estimation of the Gaussian mixture parameters $z \sim \mathcal{N}(\mu_j, \Sigma_j)$. Finally, a DNN model $f(z;\theta)$ takes z as an input with the model parameter θ. Model θ is a privacy-preserving model that can produce synthetic samples.

Algorithm 1 DeepClean: Deep Clustering Generative Model

Require: Image Data : x, # of clusters : k, Bounds on the GMM parameters $w_{min}, \sigma_{min}, \sigma_{max}$, learning parameter α, β, Privacy parameters $\varepsilon, \delta > 0$

Ensure: A Privacy-Preserving Model θ

1: $[G_1, G_2, ..., G_k] \leftarrow PGMM$

$(x, k, R, w_{min}, \sigma_{min}, \sigma_{max}, \varepsilon, \delta)$

2: **for** j from 1 to k **do**

3: $(\mu_j, \Sigma_j) \leftarrow PGE((G_j); R, w_{min}, \sigma_{min}, \sigma_{max}, \varepsilon, \delta)$*; Comment: Proof of PGE [152];*

$\pi_j \leftarrow |G_j| + 2\sqrt{2\ln(1.25/\delta)}/\varepsilon$*;*

4: **end for**

5: set weight such that for all; $j, w_j \leftarrow \pi_j/(\Sigma\pi_j)$

6: $z \leftarrow (\mu_j, \Sigma_j, w_j)_{j=1}^{k}$

7: $\hat{x} \leftarrow f(z, \theta)$

Next, let's dive into the formal analysis and justification of the version of the algorithm used to design DeepClean. To generate the synthetic data $\hat{x}$ as shown in Algorithm 1, we encode input image x by a DNN $g(z, \phi)$. The latent space z holds the Gaussian distribution of the input data, where ϕ is the encoder parameter. The GMM partitions the distribution into G_z clusters. Data object samples are similar within a cluster and are supervised to produce a labelled cluster of a set of tuples $G_\rho = ((G_1, \rho_1), \ldots., (G_z, \rho_z))$, where ρ, is a label component in the i.i.d clusters. Then inject Gaussian noise into the classified private clusters by the function DP-GMM$(G_z, \alpha, \beta, \varepsilon, \delta)$, where x is the estimator accuracy of variational distance, β is the failure parameter and the

privacy parameters ε, δ. The decoder network $f(G_z; \theta)$ generates the synthetic image output.

5.3.1 Gaussian Mixture Model (GMM)

Assuming the underlying distribution G is a mixture of k Gaussian in high-dimension d, $\{G_i \in R^d\}_{i=1}^k$ is a k distinct Gaussian distribution with dimension d. The cluster component G_i is chosen with probability $w_i \in [0,1]$, and the mean $\mu_i \in R^d$ and variance $\Sigma \in R^{d*d}$ are the parameters of the distributed Gaussian. The mixture can be written as the tuple $\{(w_i, \mu_i, \Sigma_i)\}_{i\in[k]}$. We can accurately recover the tuple $\{(\hat{w}_i, \hat{\mu}_i, \hat{\Sigma}_i)\} i \in [k]$ for a mixture $\hat{G}$. Where $\|\hat{w} - w\|_1, \|\hat{\mu}_i - \mu_i\|_{Zi}$, and $\|\hat{\Sigma} - \Sigma\|_{Zi}$ are small for every $i \in [k]$. The vector $\|.\|_Z$ approximately ensures that $\mathcal{N}(\mu_i, \Sigma_i)$ and $\mathcal{N}(\hat{\mu}_i, \hat{\Sigma}_i)$ are close in total variation distance and likewise $\|.\|_1$ ensures the same for comparing the weights.

To learn from the GMM with n samples, independent identically distributed (i.i.d.) samples can be obtained from the mixture D and roughly approximate the parameters of a mixture $\hat{D}$ by a probability $\pi : [k] \longrightarrow [k]$ and satisfying two conditions. One is a separate condition that measures the learning guarantees of the clustering and shows how the clusters are well-separated. In this case, it will ensure that privacy is adequately controlled within the clusters and limit privacy loss due to distributional assumptions. Secondly, certain boundedness of the mixture components is assumed to control the output. Let the separation condition satisfy,

$$\forall 1 \leq i < j \leq k, \|\mu_i - \mu_j\|_2 \geq s.max\sigma_i, \sigma_j$$

For $s > 0$, the Gaussian mixture $D \in G(d,k)$ is s-separated. Depending on the number of mixtures and independent of the dimension d. Assuming some large known quantities $R, \sigma_{max}, \sigma_{min}$ such that

$$\forall i \in [k] \|\mu_i\|_2 \leq Rand\sigma_{min}^2 \leq \|\Sigma_i\|_2 \leq \sigma_{max}^2$$

Definition 1- (α, β)-learning: Let the parameters of a Gaussian mixtures $D \in G(d,k)$ be $\{(\mu_1, \Sigma_1, w_1), \ldots, (\mu_k, \Sigma_k, w_k)\}$, an algorithm (α, β)-learns a distribution D and outputs a distribution $\hat{D} \in G(d,k)$ parameterized by $\{(\hat{\mu}_1, \hat{\Sigma}_1, \hat{w}_k), \ldots, (\hat{\mu}_k, \hat{\Sigma}_k, \hat{w}_k)\}$, with a probability of at least $1 - \beta$ and a permutation $\pi : [k] \longrightarrow [k]$. The following conditions will hold

1. $1 \leq i \leq k d_T v(\mathcal{N}(\mu_i, \Sigma_i), \mathcal{N}(\hat{\mu}\pi(i), (\hat{\Sigma}\pi(i)) \leq O(\alpha)$

2. $\forall 1 \leq i \leq k, |w_i - \cap w\pi(i)| \leq O(\alpha/k)$

Both conditions imply that $d_T v(D, \hat{D}) \leq \alpha$

Definition 2- Learning Labelled clusters – we learn the mixture of Gaussian, where G_i can be chosen from a mixture component $G_{\rho i}$. Such that ρ_i is a label to predict the

mixture component in the i.i.d. clusters. A labeled cluster is a set of tuples G_ρ = ((G_1, ρ_1), ..., (G_m, ρ_m)) sampled from a distribution D, where

$$D \in G(d, k, \sigma_{min}, \sigma_{max}, R, w_{min}, s)$$

The label ρ is composed of a matrix

$$\rho = \rho(i, j)$$

which is the same size as D. Each element $\rho(i, j)$ is a label of corresponding pixels in the original data X. Let p_t denote the label of sensitive clusters in G. The classification result maps of the non-sensitive clusters in the original distribution $\hat{D}$ should be similar to each other.

This analysis aims to locate the clusters distinctly so that sensitive clusters are perturbed and non-sensitive clusters are unperturbed. So, divide the image into sensitive and non-sensitive parts using masking, where M_t and M_o denote the parts, respectively. M_t is $0-1$ binary matrix which equals $M_t(i, j)$, where $M_t(i, j) = 1\ iff \rho(i, j) = \rho_t$ and $M_o = 1 - M_t$ where 1 is an all one matrix with the same size as M_t. The GMM algorithm locates the object clusters by their binary number label.

5.3.2 Variational Autoencoder Technique

In the inference process of the VAE, the encoded latent variable z is obtained from sampling the output of the Gaussian mixture $z \sim \mathcal{N}(\mu_j, \sigma_j^2)$. The reparameterisation trick is used to adapt the recognition model $q(z|G_i)$ to approximate the time posterior

distribution $p_\theta(z|G_i)$. So, make z be a deterministic function of ϕ and some noise ε, where $z = f(\phi, \varepsilon)$. A sample can be drawn from a normal distribution like $z = \mu + \sigma\varepsilon$, where $\varepsilon \sim \mathcal{N}(0, I)$.

In the generative process, the obtained latent variable z is decoded to obtain another distribution $p_\theta(z)$, where the synthetic image $\hat{x}$ can be sampled. The DNN parameters ϕ and θ are jointly learned by optimising the ELBO using the Stochastic gradient descent of the DNN. The ELBO is computed as the difference between the latent variable distribution and the observed variable distribution as follows;

$$Logp(x) \geq L(x) = Eq_\phi(z|x)[logp_\theta(x|z)] - KL(q_\phi(z|x)\|p_\theta(z)) \tag{5.1}$$

Where the first term of the difference is the expected log-likelihood, and the second is the KL divergence.

To improve the visual quality of the non-private areas, we inject information about the non-private clusters into the generative process of the decoder. The conditional information ρ' has the same size as ρ and only holds information about the non-private objects. Hence, the conditional VAE reconstruct most labelled non-private areas to preserve utility. The loss function for the conditional VAE based on the generative model is stated as

$$Lc(x) = Eq_\phi(z|x)[logp_\theta(x|z, \rho')] - KL(q_\phi(z|x)\|p_\theta(z)) \tag{5.2}$$

5.3.3 Differential Privacy

A randomised mechanism M will satisfy (ε, δ)-differential privacy $((\varepsilon, \delta) - DP)$ for learning mixtures of Gaussian if it takes two pair of image data (X, $\hat{X}$) that differ in one single item (pixel), the distributions $M(X)$ and $M(\hat{X})$ are precisely (ε, δ)-close. If the image data is partitioned into cluster distributions $X_1, \ldots, X_k \sim$ D for a mixture D satisfying separation and boundedness, $M(X)$ produces an approximate output to the parameter of G. The images $X, \hat{X} \in M$ and every set of output O, if M satisfies

$$Pr[M(X) \in O] \leq e^{\varepsilon}.Pr[M(\hat{X}) \in O] + \delta$$

Where $Pr[.]$ denotes the probability of an event, and δ bounds the probability of the privacy guarantee not holding, which is often better set to be less than $1/|D|$. Specifically, the distribution of $A(D)$ and A($\hat{D}$) are (ε, δ)-close.

Let's define the global Lp-sensitivity of the feature vector $f(x)$, as noise is injected into the cluster centres of specific locations in the image. If the images consist of n pixels, such that $X = (x_1, \ldots., x_n)$ and $\hat{X} = (\hat{x}_1, \ldots., \hat{x}_n)$, the function f maps the image to feature space, and the sensitivity $\triangle f$ is defined as

$$\triangle pf = \max_{X, \hat{X}} ||f(X) - f(\hat{X})||p$$

where X, $\hat{X}$ are neighbouring datasets, $\triangle_f$ is the maximum differences in $f(x)$ generated by two different images, and $\|.\|p$ denotes the $Lp - norm$.

The private GMM achieves differential privacy by injecting Gaussian noise, defined in the following.

Gaussian Mechanism (GM): The GM with parameter σ adds noise scaled to $\mathcal{N}(0,\sigma^2)$ to each of the private components of the output. For any $G(X) = f(X) + [N1(0, \triangle 2f.\alpha)],\ldots,$ $Nd(0, \triangle 2f.\sigma]$ where $Ni(0, \triangle 2f.\sigma)$ are i.i.d. normal random variables with zero mean and variance $(\triangle 2f.\sigma)^2$. Let $\varepsilon \in (0,1)$ be arbitrary. For $c^2 > 2\ln(1.25/\delta)$, the Gaussian mechanism with parameter $\sigma \geq c \triangle 2f/\varepsilon$ is $(\varepsilon,\delta) - DP$.

To learn the differentially private GMM with well-separated and bounded image object clusters, we describe the private GMM conditions in the following theorem (the proof is available in [152]).

Theorem 1 : A (ε,δ)-differentially private algorithm takes n samples from an unknown mixture of k Gaussians $D \in R^d$ satisfying the above conditions of separation and boundedness.

$$n = \Big(\frac{d^2}{\alpha^2 w_{min}} + \frac{d^2}{\alpha w_{min}\varepsilon} + \frac{poly(k)d^{3/2}}{w_{min}\varepsilon}\Big).poly\log\Big(\frac{dkR(\sigma_{max}/\sigma_{min}}{\alpha\beta\varepsilon\delta}\Big) \tag{5.3}$$

Where $W_{min} = min_i w_i$, with probability at least $1-\beta$, learning the parameters of D up to error α. The parameters $\alpha,\beta,\varepsilon,\delta$ are the estimator accuracy of variation distance, failure probability, and privacy parameters. R is the radius of a ball at the centre containing all means, and k is the ratio of the variances' upper and lower bounds.

Under Theorem 1, transform data to a lower dimension space and recursively cluster

the data with a Principal Component Analysis (PCA) [162]. This approach ensures the maximum effect of the injected noise. The PCA projection privately learns under the following assumptions: (i) All components being spherically Gaussian such that each component's variances lie in a small known range (with bounder ratio by a constant factor), (ii) The means of the Gaussian lie in a small ball around the origin. Making the PCA private by injecting noise into the covariance matrix makes the algorithm private. The projection shifts each component mean by the complexity of $O(\sqrt{k\sigma_{min}})$ under the already stated assumptions and preserves the separation of data because all variances are within a constant factor of one another. Finally, cluster data using the 1-cluster method of [163] and learn each component's parameters using a simplified version of [164].

Chapter 6

Experimental Results and Findings

6.1 DeepClean Performance

For DeepClean's evaluation, this study uses a dataset which is a high-dimensional street view scene from Cityscapes [165]. The image data consists of 2975 training sets, 500 validation, and 1525 test sets showing street views of different cities at different times. The images have a size of $256 * 256$ and are trained with no data augmentation because the DNN learnt more patterns and trained faster without it. Then set up the deep learning Python and Tensorflow implementation code on a Colab playbook.

Training method – The experiments follow the same setup of the VAE network in ADGAN-II [14] by setting epochs to 150 and batch-size of 1. For DeepClean, the latent dim is 128, label dim of 64, beta $\beta = 0.65$, and the learning rate of 0.001.

For the comparative analysis, this study evaluates the performance of DeepClean in comparison with two benchmark techniques for AV camera data, i.e., ADGAN [14] and VAE+DP-Kmeans [87]. These two techniques were chosen due to their balanced privacy/utility claims and their use of VAE models (similar to DeepClean). Regarding the selected dataset, ADGAN was evaluated using the Cityscapes dataset, while the VAE+DP-Kmeans model was only evaluated on MNIST and TRANSIT datasets [87].

The comparison results (provided in this section) show that DeepClean outperforms the considered benchmark techniques by preserving the better visual quality of the non-private object parts of an image while resisting location inference attacks. A brief explanation of these three techniques is provided in the following to improve readability.

- ADGAN [14] – combines VAE and GAN. The synthetic image is generated by the generator transformation $\hat{x} = G(x)$ and applies a privacy loss function $L_{pri}(G)$ to make $\hat{x}$ privacy-preserving.

- VAE + DP-kmeans [87] – combines VAE and private Kmeans. The synthetic image is generated by adding differentially private Kmeans on the data points $D = x_1, \ldots, x_N$, the results of the cluster data are produced by a DPKmeans (Parameters) = $D_1, D_2, \ldots, D_k$. The output of the parameters is used to learn the VAE generator.

- DeepClean (VAE + DP-GMM) – is the proposed method in this thesis to combine VAE and a private GMM. The GMM is applied to the latent distribution to learn sensitive and non-sensitive objects in clusters. Gaussian noise is applied to the sensitive clusters, while the noise does not impact non-sensitive objects. The clusters are then trained in a conditional VAE system.

6.1.1 Image Privacy and Utility Evaluation

To measure the efficiency of the techniques, FCN-score is adopted to quantify the features in the generated synthetic images. FCN score is efficiently adopted to evaluate generative models quantitatively [166]. Two indicators from the FCN score are used for the evaluation: pixel accuracy (PA) and interaction over union (IoU). The PA value estimates how well the image pixels are represented in percentages. In contrast, IoU value estimates the overlap between the predicted segmentation and the ground truth over the area of the union between the predicted segmentation and the ground truth. Then run the semantic segmentation model to compute the PA and IoU values of the generated images.

The evaluation of the indicators is defined as comparing performances using three metrics, i.e., image quality (IQ), image utility (IU), and image privacy (IP). IQ is estimated by taking the average PA and IoU over the whole image, IU is calculated by averaging PA and IoU over non-private objects in the image, and IP is estimated by averaging PA and IoU over the private objects in the image data. As for the metrics IQ and IU, the higher their value, the better the image representation performance of the technique. While for IP, the lower the value, the more privacy preserved and the more difficult it is to recognise an object from the image.

It is initially shown that DeepClean produces better IQ and IP than the other techniques. Table 6.1 shows the FCN-scores comparison of DeepClean with the other techniques using the Cityscapes dataset. DeepClean achieves a global IQ accuracy

of 68.30% PA and 17.15% IoU, slightly as good as ADGAN, 70.69% PA and 17.39% IoU, and VAE+DP-kmeans with 64.60% PA and 15.86% IoU. The drop in performance of DeepClean compared to ADGAN is due to achieving better privacy preservation in the private areas of the images. However, the overall IQ performance can be improved by reducing the number of noisy scales on the IP. DeepClean preserves more privacy by achieving a lower IP value, 6.36% PA and 2.76% IoU, compared to the other models. By this, DeepClean shows better resistance to privacy attacks. The goal to preserve more utility around the non-private object areas is achieved, with IU measurement of 77.75% PA and 21.20% IoU for DeepClean, which is better than the other models. The good performance of DeepClean is due to the good clustering proficiency of GMM on the distributions. However, the two deep clustering models show the effectiveness of good clustering in better controlling the image quality of specific locations in the images.

Table 6.1: The FCN-score comparisons of various generative models on the cityscape dataset.

Model	**Image Quality(IQ)**		**Image Privacy(IP)**		**Image Utility(IU)**	
	PA	**IoU**	**PA**	**IoU**	**PA**	**IoU**
ADGAN [14]	70.69%	17.39%	11.65%	4.72%	77.53%	21.06%
VAE+DP-Kmeans [87]	64.60%	15.86%	6.27%	2.37%	60.54%	16.53%
DeepClean	68.30%	17.15%	6.35%	2.76%	77.58%	23.04%

Figure 6.1 shows the accuracy of the clustering technique over some epochs in training the Cityscapes dataset. The number of clusters k was initially set to 10 to achieve high clustering performance. For the privacy parameters for the benchmark techniques, the

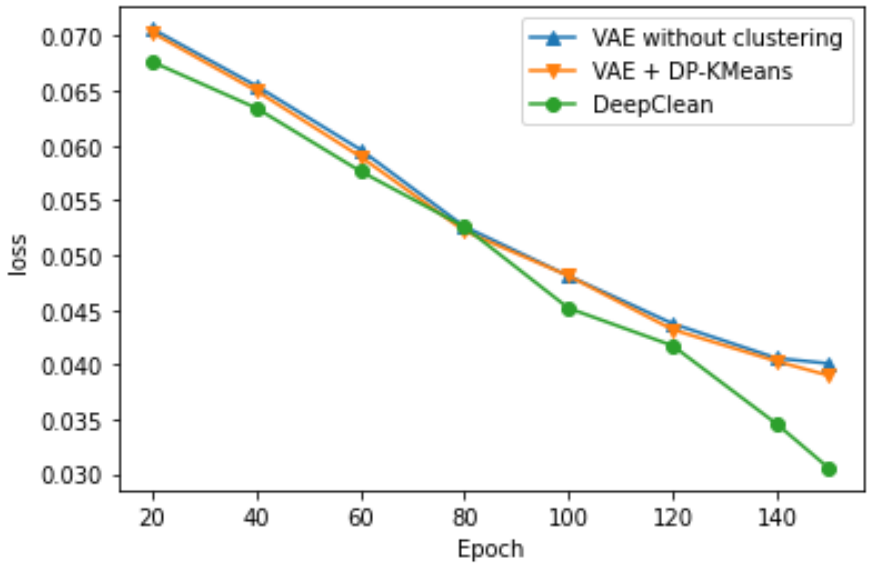


Figure 6.1: Clustering accuracy over some epochs during training on the Cityscapes dataset.

default settings in the K-means model [87] are used, and for the clustering models, noise scales for clustering σ_k are set as 1.0 and SGD noise scale σ_G as 40. The metric result shows that the DeepClean model achieves reasonable privacy protection better than ADGAN concerning the utility gained in the non-private object areas.

6.1.2 Privacy Performance

To validate the privacy protection performance achieved by the proposed technique, run the geo-localisation attack using the LIFT geo-localisation technique to localise the query image data. The reference dataset used for the experiment is 102k google street view images covering European cities. Then select 500 sets from Section 4 Cityscape test set for the query image set. LIFT quantifies the percentages of images that can be localised at 300m from their actual locations. Localisation above the 300m

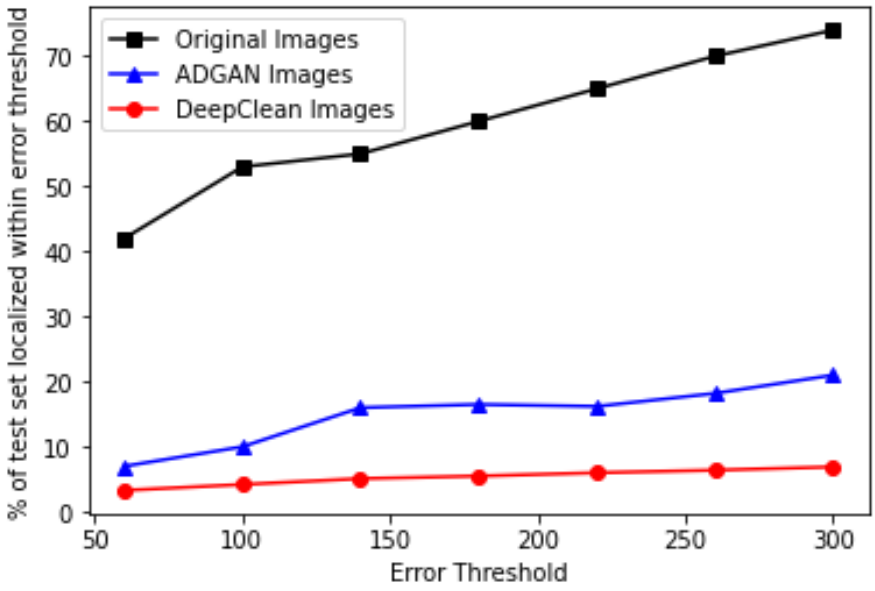


Figure 6.2: Privacy performance of DeepClean compared with other techniques.

range is regarded as a non-matching nearest neighbour. Using the LIFT technique and the voting scheme step for feature matching and geolocating the best matching reference image, respectively, improves the performance of geo-localisation than the Multi-KNN approach used in other studies for privacy performance.

Figure 6.2 shows the privacy performance of DeepClean on the images compared with the benchmark studies. The X-axis is the error threshold in meters, and Y-axis is the percentage of the test set localised within the error threshold. LIFT localises the original query images at 74%, about 300m better than Multi-KNN 60%. The higher percentage result proves a higher risk of location inference threats on the image data. On the other hand, using distorted images of ADGAN models as the query image, localisation improves from 5% to 20% within the error threshold of 60m – 300m. This improvement indicates that DSC can still match some of its features to the produced

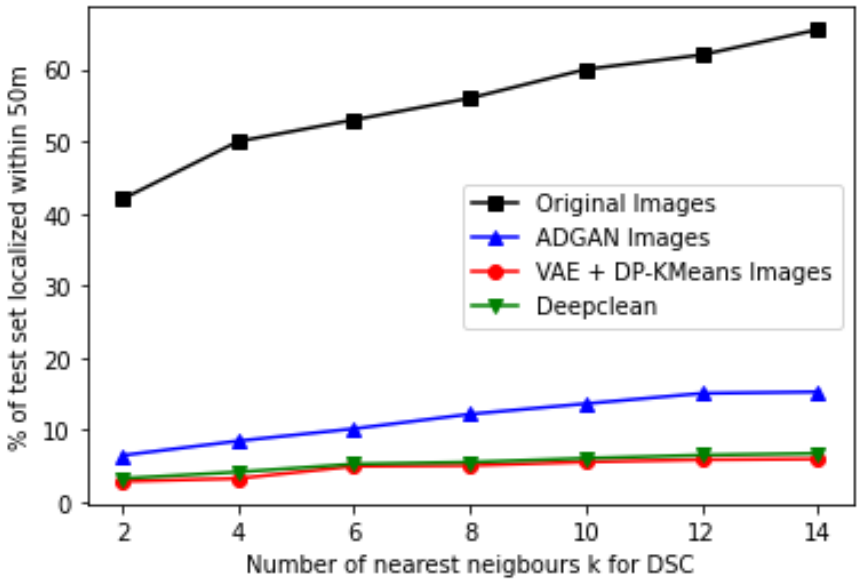


Figure 6.3: Performance comparisons of the techniques with fixed nearest neighbour.

dynamic corresponding reference data set. DeepClean reduces localisation accuracy to about 3% - 7%, which is relatively minimal compared to the other techniques. With this result, there are possibilities that the original reference images are not included among the matching nearest neighbour images. Both local and global features present around the classified private object areas are well distorted to confuse the LIFT technique from detecting stable features. Only a few images with more stable features around features such as road signs, vegetation and structures, apart from buildings, likely make the matching step. However, the image is unlikely to return as the best matching image. This result makes Deepclean images immune to location inference attacks.

As seen in Figure 6.3, this analysis tested a fixed Multi-KNN to examine the performance of the DSC on different numbers of nearest neighbours. Although Multi-KNN used in previous works drops in performance when k is ≥ 4, DSC also improves the

Table 6.3: SSIM measurement on Cityscapes dataset.

Model	SSIM Measurement
ADGAN	0.6210
VAE+DP-Kmeans	0.4560
DeepClean	0.6012

chances of selecting the original image data as the nearest neighbour increases. It localises the data at an accuracy of 40% to 65% within an error threshold of 50m. The first 4 NNs retrieved by the multi-KNN method assume the NNs are the stable features detected from the image. DSC detected features show that they contribute more to the localisation accuracy. However, DeepClean images localise at around 4% to 6%. These results show that images generated by DeepClean are immune to these attacks.

6.1.3 Utility Performance

For evaluating the DeepClean model's utility performance, SSIM (structural similarity index) of the generated images is measured. SSIM measures image recognition utility very close to human visibility [161]. It measures the original and distorted data similarity by a number greater or equal to 0 and less or equal to 1, where 0 means completely different, and 1 means the same. Table 6.3 shows that DeepClean achieves 0.6012 on the Cityscape data, which is closer to the value achieved by ADGAN. The slight drop in utility performance of DeepClean compared to ADGAN considers the

stricter privacy requirements enforced in the private object areas. This highlights the challenge of simultaneously achieving a balanced privacy-utility trade-off in images. Thus, the privacy-utility performance results show that a balanced trade-off may not be achievable to suit all requirements. Therefore, it explains my approach to achieving more utility in the non-private object areas. The results produced by DeepClean, as shown in Figure 6.4 (among other image data generated by the other techniques), generate a more balanced privacy-utility trade-off regarding more privacy preservation in private object areas and utility preserved in non-private object areas. DeepClean generated data can be used to train AV driving navigation models.

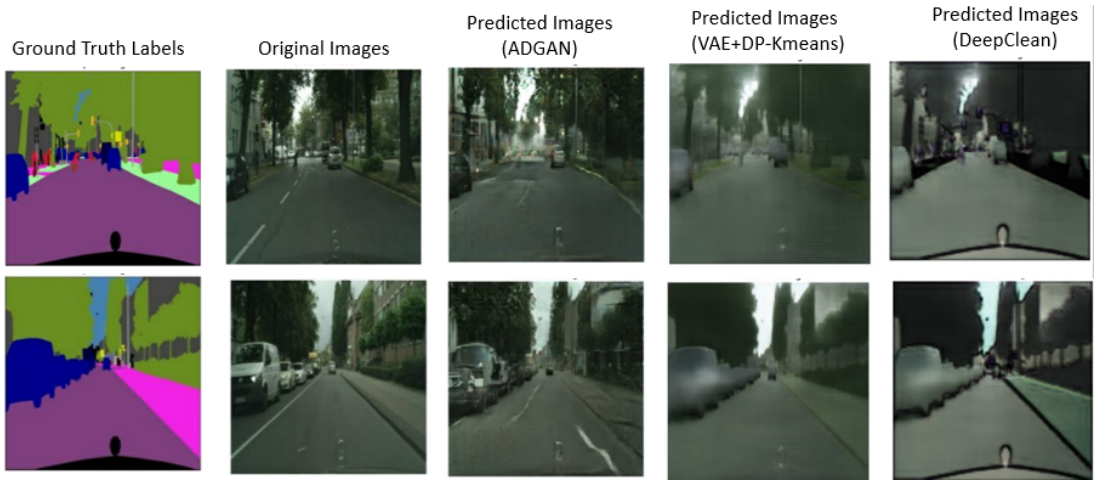


Figure 6.4: Visual quality of non-sensitive object areas and privacy-preservation of sensitive object areas comparisons of the techniques on Cityscapes data.

Chapter 7

Conclusion and Future Work

7.1 Thesis Summary

This thesis analyses the impact of location inference attacks on AV camera data and develops a privacy-preserving method employing a privacy-amenable generative model to account for privacy. The approach aims to reduce the attack's impact and simultaneously preserve sufficient data utility for analysis. Traditional privacy-preserving techniques, such as blurring, pixelating, etc., are inefficient in handling the privacy/utility requirements for AV camera data storage and processing. The output of such techniques is either too blurry to be used for further analysis or less blurry and prone to a location inference attack. Recent approaches design private generative models to reconstruct camera data for achieving balanced privacy/utility requirements. A state-of-the-art approach to enforcing the privacy/utility requirements adopts GAN-based models without provable privacy guarantees. Their privacy implementation involves simple distance measurement and reconstruction loss. In this thesis, a developed location inference attack (LIFT) technique exploited the location privacy of GAN-based generated camera data by learning with tailored noisy images as training data and recognising object features in the images using their discriminative attributes. The location inference tool, LIFT, is developed to attack the location privacy of the dataset.

It defines a novel distinctive nearest neighbour selection process using a robust similarity metric to arrange stable features at the top of the graph and less stable features at the rear of the graph for a robust matching process. The similarity metric construction used to learn the discriminative features was adopted from the work of Pavan and Pelillo, consisting of a Gaussian Kernel measuring the similarity of some feature representation of the pixels. Feature matching is performed using a formulated robust pairwise clustering technique to handle large feature selection and matching as graph cliques. Then the best matching reference is chosen using a simple voting scheme. LIFT was evaluated using a dataset from Google Street View and the Berkeley repository. The feature-matching performance of LIFT compared with other state-of-the-art geo-localisation tools shows that LIFT could match features of some of the ADGAN images even though with few features detected and fewer false positive results. LIFT outperforms the benchmark techniques by 20% in terms of the overall geo-localisation. This result means that LIFT geolocalised the query images with the corresponding reference images enough to locate places and link rough trajectories. This result also points out the significance of location inference threats on AV camera data and the need to improve the privacy preservation of AV data for long-term storage and data processing. Lastly, the result proposes robust access control implementations, authentication systems and data sharing for the AV to eliminate or mitigate unauthorised access and potential AV data misuse.

This thesis addresses the challenge of a balanced privacy/utility trade-off for AV camera data by designing DeepClean, a deep learning technique to generate efficient privacy-

preserving synthetic AV camera data. The main contribution of this thesis, achieved through DeepClean, is to develop a privacy-amenable generative model that combines a Gaussian mixture model with a Variational Autoencoder to learn distinct cluster structures. DeepClean's implementation is two-fold- the first is to learn well-separated object clusters and classify them as private and non-private. Then, inject noises into private object clusters without affecting many underlying structures of the non-private object clusters. Secondly, the cluster outputs are reconstructed using a powerful VAE architecture to produce high-dimensional data. This approach preserves guaranteed privacy around the sensitive object parts to resist LIFT's localisation. The resulting generative model guarantees differential privacy and resists a robust location inference attack (such as LIFT and GMCP) by less than 4% localisation accuracy, which implies that it is less likely to localise a subject using a robust attack model. The overall image utility level is reasonably comparable to the benchmark studies. Conclusively, the generated data from the model are suitable for long-term storage and data processing.

7.2 Conclusion

Location inference attacks threaten the privacy of Autonomous vehicle camera data. For this reason, a reasonable level of security and privacy is required to enhance data storage and sensitive image protections, respectively. Focusing on the privacy-preservation of AV camera data, this thesis has addressed the privacy/utility trade-off for efficient data analysis and storage. The developed generative model approach integrates a differentially private technique to guarantee privacy instead of relying on

masking or reconstruction loss for privacy protection by prior works. The experimental analysis of the models showed that the developed technique achieved more privacy preservation and comparable utility performance to benchmark models. Hence, the conclusion was drawn that DeepClean-generated images benefit privacy-preserving long-term storage and data processing.

7.3 Future work

The observed limitation of the developed location inference attack (LIFT) technique assumes that a query image has a matching reference in the database. However, there can be no match if that is not the case. Also, LIFT will produce low localisation accuracy on too blurry query images. The result may be false positive in some cases where image objects of a specific location are very similar to those of another location, mainly occurring in city-wide image data. Future direction on location inference attacks on AV camera data could adopt and train a convolutional neural network to perform similar phases as LIFT to achieve better geo-localisation performance.

On the other hand, the practical limitation of DeepClean is that an image with many similar objects, e.g., buildings, dominating most of the image may need to be better reconstructed. In other words, if a large part of the image is sensitive and with added noise, the whole image may sometimes be rendered blurry. Thus, DeepClean overall utility could be improved. However, privacy guarantees must simultaneously be achieved. Future direction on AV camera data privacy could formulate a GAN-based model amenable to differential privacy, which aims to utilise generative and discrimi-

native models for an improved image utility with a provable privacy guarantee.

REFERENCES

[1] CCAV, "Uk connected autonomous vehicle research development projects 2018," p. 64, 2017.

[2] V. K. Veitas and S. Delaere, "In-vehicle data recording, storage and access management in autonomous vehicles," 2018.

[3] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, "Bdd100k: A diverse driving dataset for heterogeneous multitask learning," Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 2633–2642, 2020.

[4] X. Li, L. Ding, L. Wang, and F. Cao, "Fpga accelerates deep residual learning for image recognition," pp. 837–840, 2017.

[5] H. C. Shin, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, and R. M. Summers, "Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning," IEEE Transactions on Medical Imaging, vol. 35, pp. 1285–1298, 2016.

[6] W. Zhang, C. Witharana, W. Li, C. Zhang, X. Li, and J. Parent, "Using deep learning to identify utility poles with crossarms and estimate their locations from google street view images," Sensors (Switzerland), vol. 18, 8 2018.

[7] A. Reich, N. A. Kramer, and R. Lenninger, “Vehicle data management a standardized access as the basis of new business models,” ATZelektronik worldwide, vol. 13, pp. 38–43, 2018.

[8] B. Martens and F. Mueller-Langer, “Access to digital car data and competition in aftersales services,” SSRN Electronic Journal, 2018.

[9] V. Dhar, “Equity, safety, and privacy in the autonomous vehicle era,” Computer, vol. 49, pp. 80–83, 2016.

[10] J. Hays and A. A. Efros, Large-scale image geolocalization. Springer International Publishing, 1 2015.

[11] E. Zemene, Y. T. Tesfaye, H. Idrees, A. Prati, M. Pelillo, and M. Shah, “Largescale image geo-localization using dominant sets,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 41, pp. 148–161, 1 2019.

[12] G. Schindler, M. Brown, and R. Szeliski, City-Scale Location Recognition.

[13] A. R. Zamir and M. Shah, “Image geo-localization based on multiplenearest neighbor feature matching using generalized graphs,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 36, pp. 1546–1558, 2014.

[14] Z. Xiong, Z. Cai, Q. Han, A. Alrawais, and W. Li, “Adgan: Protect your location privacy in camera data of auto-driving vehicles,” IEEE Transactions on Industrial Informatics, vol. 17, pp. 6200–6210, 9 2021.

[15] Z. Xiong, W. Li, Q. Han, and Z. Cai, "Privacy-preserving auto-driving: A gan-based approach to protect vehicular camera data," vol. 2019-November, pp. 668–677, Institute of Electrical and Electronics Engineers Inc., 11 2019.

[16] T. Sattler, M. Havlena, K. Schindler, and M. Pollefeys, "Large-scale location recognition and the geometric burstiness problem," vol. 2016-December, pp. 1582–1590, IEEE Computer Society, 12 2016.

[17] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "Netvlad: Cnn architecture for weakly supervised place recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 40, pp. 1437–1451, 6 2018.

[18] Z. Xiong, W. Li, Q. Han, and Z. Cai, "Privacy-preserving auto-driving: A gan-based approach to protect vehicular camera data," Proceedings - IEEE International Conference on Data Mining, ICDM, vol. 2019-November, pp. 668–677, 2019.

[19] J. Petit and S. E. Shladover, "Potential cyberattacks on automated vehicles," IEEE Transactions on Intelligent Transportation Systems, vol. 16, pp. 546–556, 2015.

[20] L. Collingwood, "Privacy implications and liability issues of autonomous vehicles," Information and Communications Technology Law, vol. 26, pp. 32–45, 2017.

[21] B. Bingham, B. Foley, H. Singh, R. Camilli, K. Delaporta, R. Eustice, A. Mallios, D. Mindell, C. Roman, and D. Sakellariou, "Brian bingham," vol. 25, pp. 1–16, 2010.

[22] R. Coppola and M. Morisio, "Connected car: Technologies, issues, future trends," ACM Computing Surveys, vol. 49, pp. 1–36, 2016.

[23] D. Harris, "Engineering psychology and cognitive ergonomics," Engineering Psychology and Cognitive Ergonomics, pp. 339–362, 2017.

[24] M. Burgess, "When does a car become truly autonomous? levels of self-driving technology explained.," WIRED, 2017.

[25] J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J. Z. Kolter, D. Langer, O. Pink, V. Pratt, M. Sokolsky, G. Stanek, D. Stavens, A. Teichman, M. Werling, and S. Thrun, "Towards fully autonomous driving: Systems and algorithms," IEEE Intelligent Vehicles Symposium, Proceedings, pp. 163–168, 2011.

[26] T. S. Catapult, "Market forecast for cav report final," 2017.

[27] D. Snl, F. Extra, and N. Y. May, "Industry reps : Insurers need broad access to data to underwrite driverless cars," pp. 1–3, 2018.

[28] A. Bloomberg, W. Service, N. York, and N. York, "Teslas don't have black boxes , making crash probes harder," pp. 1–4, 2018.

[29] R. Brunauer and K. Rehrl, "Supporting road maintenance with in-vehicle data: Results from a field trial on road surface condition monitoring," IEEE Confer-

ence on Intelligent Transportation Systems, Proceedings, ITSC, pp. 2236–2241, 2016.

[30] P. Sun, H. Kretzschmar, X. Dotiwalla, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, V. Vasudevan, W. Han, J. Ngiam, H. Zhao, A. Timofeev, S. Ettinger, M. Krivokon, A. Gao, A. Joshi, J. Shlens, Z. Chen, D. Anguelov, and W. Llc, "Scalability in perception for autonomous driving : Waymo open dataset,"

[31] P. Nelson, "Just one autonomous car will use 4000gb of data/day.," Network World, p. 1, 2018.

[32] Navya, "Navya safety report," NAVYA safety report, vol. 1.1, p. 43, 2019.

[33] E. C. D. Move, "Access to in-vehicle data and resources," 2017.

[34] M. Kyriakidis, R. Happee, and J. C. F. D. Winter, "Supplementary materials for public opinion on automated driving: Results of an international questionnaire among 5000 respondents," vol. 32, pp. 127–140, 2015.

[35] D. J. Fagnant and K. Kockelman, "Preparing a nation for autonomous vehicles: Opportunities, barriers and policy recommendations," Transportation Research Part A: Policy and Practice, vol. 77, pp. 167–181, 2015.

[36] Q. Wu, J. Domingo-Ferrer, and Ursula Gonzalez-Nicolas, "Balanced trustworthiness, safety, and privacy in vehicle-to-vehicle communications," IEEE Transactions on Vehicular Technology, vol. 59, pp. 559–573, 2010.

[37] S. Plosz and P. Varga, "Security and safety risk analysis of vision guided autonomous vehicles," 2018 IEEE Industrial Cyber-Physical Systems (ICPS), pp. 193–198, 2018.

[38] C. Owensby, M. Tomitsch, and C. Parker, "A framework for designing interactions between pedestrians and driverless cars: Insights from a ride-sharing design study," ACM International Conference Proceeding Series, pp. 359–363, 2018.

[39] M. Taraba, J. Adamec, M. Danko, and P. Drgona, "Utilization of modern sensors in autonomous vehicles," 2018 ELEKTRO, pp. 1–5, 2018.

[40] F. Kunz, D. Nuss, J. Wiest, H. Deusch, S. Reuter, F. Gritschneder, A. Scheel, M. Stubler, M. Bach, P. Hatzelmann, C. Wild, and K. Dietmayer, "Autonomous driving at ulm university: A modular, robust, and sensor-independent fusion approach," IEEE Intelligent Vehicles Symposium, Proceedings, vol. 2015-Augus, pp. 666–673, 2015.

[41] L. Butcher, "Vehicle cybersecurity framework developed by fev," Automotive Testing Technology International, 2021.

[42] M. K. K. Kishore and M. Suvitha, "An efficient pseudonymous authentication scheme with strong privacy," International Journal of Emerging Trends in Science and Technology, vol. 2, pp. 1792–1799, 2015.

[43] U. Rajput, F. Abbas, and H. Oh, "A hierarchical privacy preserving pseudonymous authentication protocol for vanet," IEEE Access, vol. 4, pp. 7770–7784, 2016.

[44] L. Benarous and A. Vanet, "Ensuring privacy and authentication for v2v resource sharing," pp. 1–6, 2017.

[45] S. F. Tzeng, S. J. Horng, T. Li, X.Wang, P. H. Huang, and M. K. Khan, "Enhancing security and privacy for identity-based batch verification scheme in vanets," IEEE Transactions on Vehicular Technology, vol. 66, pp. 3235–3248, 2017.

[46] M. M. Islam, A. Lautenbach, C. Sandberg, and T. Olovsson, "A risk assessment framework for automotive embedded systems," pp. 3–14, 2016.

[47] D. Dominic, S. Chhawri, R. M. Eustice, D. Ma, and A. Weimerskirch, "Risk assessment for cooperative automated driving," pp. 47–58, 2016.

[48] CNIL, "Connected vehicles and personal and personal data summary," 2017.

[49] ICO, "Guide to the general data protection regulation (gdpr)," Information Commissioner's Office.

[50] M. Deng, K. Wuyts, R. Scandariato, B. Preneel, and W. Joosen, "A privacy threat analysis framework: supporting the elicitation and fulfillment of privacy requirements,"

[51] A. Mehmood, I. Natgunanathan, Y. Xiang, G. Hua, and S. Guo, "Protection of big data privacy," IEEE Access, vol. 4, pp. 1821–1834, 2016.

[52] Y. A. A. S. Aldeen, M. Salleh, and M. A. Razzaque, "A comprehensive review on privacy preserving data mining," SpringerPlus, vol. 4, pp. 1–36, 2015.

[53] M. U. Hassan, M. H. Rehmani, and J. Chen, "Differential privacy techniques for cyber physical systems: A survey," IEEE Communications Surveys and Tutorials, vol. 22, pp. 746–789, 2020.

[54] J. Soria-Comas, J. Domingo-Ferrer, D. Sanchez, and D. Megias, "Individual differential privacy: A utility-preserving formulation of differential privacy guarantees," IEEE Transactions on Information Forensics and Security, vol. 12, pp. 1418–1429, 2017.

[55] L. Chen, S. Thombre, K. Jarvinen, E. S. Lohan, A. Alen-Savikko, H. Leppakoski, M. Z. H. Bhuiyan, S. Bu-Pasha, G. N. Ferrara, S. Honkala, J. Lindqvist, L. Ruotsalainen, P. Korpisaari, and H. Kuusniemi, "Robustness, security and privacy in location-based services for future iot: A survey," IEEE Access, vol. 5, pp. 8956–8977, 2017.

[56] J. Domingo-Ferrer, J. Soria-Comas, and R. Mulero-Vellido, "Steered microaggregation as a unified primitive to anonymize data sets and data streams," IEEE Transactions on Information Forensics and Security, vol. 14, pp. 3298–3311, 2019.

[57] S. Parkinson, P. Ward, K. Wilson, and J. Miller, "Cyber threats facing autonomous and connected vehicles: Future challenges," IEEE Transactions on Intelligent Transportation Systems, vol. 18, pp. 2898–2915, 2017.

[58] A. Cavoukian, "International council on global privacy and security, by design," IEEE Potentials, vol. 35, pp. 43–46, 2016.

[59] C. Paul Pittman Steven R, "Ccpa and gdpr: Comparison of certain provisions," White Case, 2018.

[60] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," 12 2013.

[61] A. Srivastava and C. Sutton, "Autoencoding variational inference for topic models," 3 2017.

[62] E. Nalisnick and P. Smyth, "Stick-breaking variational autoencoders," 5 2016.

[63] H. Zhang, B. Chen, D. Guo, and M. Zhou, "Whai: Weibull hybrid autoencoding inference for deep topic modeling," arXiv, pp. 1–15, 2018.

[64] W. Joo, W. Lee, S. Park, and I. C. Moon, "Dirichlet variational autoencoder," Pattern Recognition, vol. 107, 2020.

[65] S. Cycles, "Chapter 9 chapter 9," Cycle, vol. 1897, pp. 44–45, 1989.

[66] C. K. Reddy and B. Vinzamuri, "A survey of partitional and hierarchical clustering algorithms," Data Clustering, pp. 87–110, 2019.

[67] A. Bouguettaya, Q. Yu, X. Liu, X. Zhou, and A. Song, "Efficient agglomerative hierarchical clustering," Expert Systems with Applications, vol. 42, pp. 2785–2797, 2015.

[68] T. Matching and R. Test, "Chapter 8" Test, vol. 1937, pp. 162–173, 2001.

[69] B. Yang, X. Fu, N. D. Sidiropoulos, and M. Hong, "Towards k-means-friendly spaces: Simultaneous deep learning and clustering," 34th International Conference on Machine Learning, ICML 2017, vol. 8, pp. 5888–5901, 2017.

[70] M. E. Celebi, Partitional clustering algorithms. 2015.

[71] R. Chitta, R. Jin, and A. K. Jain, "Efficient kernel clustering using random fourier features," pp. 161–170, 2012.

[72] M. M. Fard, T. Thonet, and E. Gaussier, "Deep k-means: Jointly clustering with k-means and learning representations," Pattern Recognition Letters, vol. 138, pp. 185–192, 2020.

[73] Z. Jiang, Y. Zheng, H. Tan, B. Tang, and H. Zhou, "Variational deep embedding: An unsupervised generative approach to clustering," IJCAI International Joint Conference on Artificial Intelligence, vol. 0, pp. 1965–1972, 2017.

[74] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 11 2015.

[75] G. Liu, J. Si, Y. Hu, and S. Li, "Photographic image synthesis with improved u-net," pp. 402–407, Institute of Electrical and Electronics Engineers Inc., 6 2018.

[76] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, pp. 640–651, 2017.

[77] P. Baldi, “Autoencoders, unsupervised learning, and deep architectures,” ICML Unsupervised and Transfer Learning, pp. 37–50, 2012.

[78] C. C. Hsu and C.W. Lin, “Cnn-based joint clustering and representation learning with feature drift compensation for large-scale image data,” IEEE Transactions on Multimedia, vol. 20, pp. 421–429, 2 2018.

[79] W. Zhou and Q. Zhou, “Deep embedded clustering with adversarial distribution adaptation,” IEEE Access, vol. 7, pp. 113801–113809, 2019.

[80] J. Xie, R. Girshick, and A. Farhadi, “Unsupervised deep embedding for clustering analysis,” 33rd International Conference on Machine Learning, ICML 2016, vol. 1, pp. 740–749, 2016.

[81] J. Enguehard, P. O’Halloran, and A. Gholipour, “Semi-supervised learning with deep embedded clustering for image classification and segmentation,” IEEE Access, vol. 7, pp. 11093–11104, 2019.

[82] M. Amiri-zarandi, R. A. Dara, and E. Fraser, “Computers and security: a survey of machine learning-based solutions to protect privacy in the internet of things,” Computers Security, vol. 96, p. 101921, 2020.

[83] Z. Wang, S. Chang, J. Zhou, M. Wang, and T. S. Huang, “Learning a taskspecific deep architecture for clustering,” 9 2015.

[84] J. Yang, D. Parikh, D. Batra, and V. Tech, “Joint unsupervised learning of deep representations and image clusters,” 2016.

[85] M.-Y. Liu, T. Breuel, and J. Kautz, "Unsupervised image-to-image translation networks," 3 2017.

[86] K. Lata, M. Dave, and K. N. Nishanth, "Image-to-image translation using generative adversarial network," pp. 2019–2022, 2019.

[87] G. Acs, L. Melis, C. Castelluccia, and E. D. Cristofaro, "Differentially private mixture of generative neural networks," IEEE Transactions on Knowledge and Data Engineering, vol. 31, pp. 1109–1121, 6 2019.

[88] J. Abowd, L. Alvisi, C. Dwork, S. Kannan, A. Machanavajjhala, and J. Reiter, "Privacy-preserving data analysis for the federal statistical agencies," pp. 1–7, 2017.

[89] S. Samarah, M. G. A. Zamil, A. F. Aleroud, M. Rawashdeh, M. F. Alhamid, and A. Alamri, "An efficient activity recognition framework: Toward privacysensitive health data sensing," IEEE Access, vol. 5, pp. 3848–3859, 2017.

[90] A. van Looy, "Social media management: technologies and strategies for creating business value," Springer texts in business and economics, p. 250, 2016.

[91] P. Dar, "Berkeley open sources largest self-driving dataset every data scientist should download now.," 2018.

[92] "Level 5 open data.," Lyft.

[93] A. Kumar and M. Gyanchandani, "A comparative survey on privacy preservation and privacy measuring techniques in data publishing," Proceedings of the

2nd International Conference on Intelligent Computing and Control Systems, ICICCS 2018, pp. 1902–1906, 2019.

[94] W. J. Kohler and A. Colbert-Taylor, "Current law and potential legal issues pertaining to automated, autonomous and connected cars," Santa Clara Computer High Tech. Law Journal, vol. 31, pp. 100 –138, 2015.

[95] A. Tockar, "Passengers privacy in the nyc taxicab dataset.," Riding with the stars, 2014.

[96] J. E. Smith, A. R. Clark, A. T. Staggemeier, and M. C. Serpell, "Disclosure control," IEEE Transactions on Evolutionary Computation, vol. 16, pp. 1–11, 2011.

[97] A. Srivastava and G. Geethakumari, "Determining privacy utility trade-off for online social network data publishing," 12th IEEE International Conference Electronics, Energy, Environment, Communication, Computer, Control: (E3-C3), INDICON 2015, pp. 1–6, 2016.

[98] S. J. De and D. L. Metayer, "Privacy risk analysis," Synthesis Lectures on Information Security, Privacy, and Trust, vol. 81, pp. 1–133, 2016.

[99] S. H. Begum and F. Nausheen, "A comparative analysis of differential privacy vs other privacy mechanisms for big data," Proceedings of the 2nd International Conference on Inventive Systems and Control, ICISC 2018, pp. 512–516, 2018.

[100] Y. Zhao, J. Wang, Y. Luo, and J. Le, "(α,β,k)-anonymity: An effective privacy preserving model for databases," Proceedings of the International Symposium on Test and Measurement, vol. 1, pp. 412–415, 2009.

[101] P. Sui, X. Li, and Y. Bai, "A study of enhancing privacy for intelligent transportation systems: K-correlation privacy model against moving preference attacks for location trajectory data," IEEE Access, vol. 5, pp. 24555–24567, 2017.

[102] J. Domingo-Ferrer and V. Torra, "A critique of k-anonymity and some of its enhancements," ARES 2008 - 3rd International Conference on Availability, Security, and Reliability, Proceedings, pp. 990–993, 2008.

[103] K. V and T. K.P, "Protecting privacy when disclosing information: K anonymity and its enforcement through suppression," International Journal of Computing Algorithm, vol. 001, pp. 19–22, 2012.

[104] T. Li, N. Li, J. Zhang, and I. Molloy, "Slicing: A new approach for privacy preserving data publishing," IEEE Transactions on Knowledge and Data Engineering, vol. 24, pp. 561–574, 2012.

[105] J. Soria-Comas and J. Domingo-Ferrert, "Differential privacy via t-closeness in data publishing," 2013 11th Annual Conference on Privacy, Security and Trust, PST 2013, pp. 27–35, 2013. 131

[106] N. Li, T. Li, and S. Venkatasubramanian, "Closeness: A new privacy measure for data publishing," IEEE Transactions on Knowledge and Data Engineering, vol. 22, pp. 943–956, 2010.

[107] J. Jia and N. Z. Gong, "Defending against machine learning based inference attacks via adversarial examples: Opportunities and challenges," pp. 1–20, 2019.

[108] A. Shaikh and S. Patil, "A survey on privacy enhanced role based data aggregation via differential privacy," 2018 International Conference On Advances in Communication and Computing Technology, ICACCT 2018, pp. 285–290, 2018.

[109] R. Assam and T. Seidl, "A model for context-aware location identity preservation using differential privacy," Proceedings - 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, TrustCom 2013, pp. 346–353, 2013.

[110] B. Yang, I. Sato, and H. Nakagawa, "Bayesian differential privacy on correlated data," Proceedings of the ACM SIGMOD International Conference on Management of Data, vol. 2015-May, pp. 747–762, 2015.

[111] P. Zhao, G. Zhang, S. Wan, G. Liu, and T. Umer, "A survey of local differential privacy for securing internet of vehicles," Journal of Supercomputing, 2019.

[112] S. Fletcher and M. Z. Islam, "Decision tree classification with differential privacy: A survey," ACM Computing Surveys, vol. 52, 2019.

[113] B. I. P. Rubinstein, P. L. Bartlett, L. Huang, and N. Taft, "Learning in a large function space: Privacy-preserving mechanisms for svm learning," Journal of Privacy and Confidentiality, vol. 4, pp. 1–21, 2012.

[114] H. C. Shin, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, and R. M. Summers, "Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning," IEEE Transactions on Medical Imaging, vol. 35, pp. 1285–1298, 2016.

[115] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning. bt - proceedings of the 22nd acm sigsac conference on computer and communications security, denver, co, usa, october 12-16, 2015," Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security - CCS '15, pp. 1310–1321, 2015.

[116] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," Proceedings of the ACM Conference on Computer and Communications Security, vol. 2015-Octob, pp. 1322–1333, 2015.

[117] P. Li, T. Li, H. Ye, J. Li, X. Chen, and Y. Xiang, "Privacy-preserving machine learning with multiple data providers," Future Generation Computer Systems, vol. 87, pp. 341–350, 2018.

[118] M. C. Lee, R. Mitra, E. Lazaridis, A. C. Lai, Y. K. Goh, and W. S. Yap, "Data privacy preserving scheme using generalised linear models," Computers and Security, vol. 69, pp. 142–154, 2017.

[119] B. Talebi and M. N. Dehkordi, "Sensitive association rules hiding using electromagnetic field optimization algorithm," Expert Systems with Applications, vol.

114, pp. 155–172, 2018.

[120] A. Multiple, “Synthetic data: An introduction 10 tools,” 2020.

[121] I. Kaloskampus, “Synthetic data for public good, data science campus, gov.uk.,”

[122] M. E. Gursoy, L. Liu, S. Truex, and L. Yu, “Differentially private and utility preserving publication of trajectory data,” IEEE Transactions on Mobile Computing, vol. 18, pp. 2315–2329, 2018.

[123] Y. H. Chen, H. H. Chen, and P. C. Huang, “Enhancing the data privacy for public data lakes,” Proceedings of 4th IEEE International Conference on Applied System Innovation 2018, ICASI 2018, pp. 1065–1068, 2018.

[124] V. K. Veitas and S. Delaere, “In-vehicle data recording, storage and access management in autonomous vehicles,” 2018.

[125] CNIL, “Connected vehicles and personal and personal data summary,” 2017.

[126] E. C. D. Move, “Access to in-vehicle data and resources,” 2017.

[127] J. Hays and A. A. Efros, “Im2gps: estimating geographic information from a single image,”

[128] J. Brejcha and M. C adık, “State-of-the-art in visual geo-localization,” Pattern Analysis and Applications, vol. 20, pp. 613–637, 8 2017.

[129] J. Hays and A. A. Efros, “Large-scale image geolocalization,” Multimodal Location Estimation of Videos and Images, pp. 41–62, 1 2015.

[130] P. Gronat, J. Sivic, G. Obozinski, and T. Pajdla, "Learning and calibrating per-location classifiers for visual place recognition," International Journal of Computer Vision, vol. 118, pp. 319–336, 7 2016.

[131] S. Se, D. G. Lowe, and J. J. Little, "Vision-based global localization and mapping for mobile robots," IEEE Transactions on Robotics, vol. 21, pp. 364–375, 6 2005.

[132] Y. Kalantidis, G. Tolias, Y. Avrithis, M. Phinikettos, E. Spyrou, P. Mylonas, and S. Kollias, "Viral: Visual image retrieval and localization," Multimedia Tools and Applications, vol. 51, pp. 555–592, 1 2011.

[133] C. Valgren and A. J. Lilienthal, "Sift, surf seasons: Appearance-based longterm localization in outdoor environments,"

[134] A. Feryanto and I. Supriana, "Location recognition using detected objects in an image," 2011.

[135] D. Lowe, "Distinctive image feature from scale-invariant keypoints," International Journal of Computer Vision, 2004.

[136] T.Weyand, I. Kostrikov, and J. Philbin, "PlaNet - photo geolocation with convolutional neural networks," 2 2016.

[137] E. Karami, S. Prasad, and M. Shehata, "Image matching using sift, surf, brief and orb: Performance comparison for distorted images,"

[138] Z. Huo, X. Meng, H. Hu, and Y. Huang, "You can walk alone : Trajectory privacy-preserving," vol. 7238, pp. 351–366, 2012.

[139] M. E. Andres, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, "Geoindistinguishability: Differential privacy for location-based systems," Proceedings of the ACM Conference on Computer and Communications Security, pp. 901–914, 2013.

[140] H. Hu, J. Xu, S. T. On, J. Du, and J. K. Y. Ng, "Privacy-aware location data publishing," ACM Transactions on Database Systems, vol. 35, 2010.

[141] F. T. Brito, C. F. Costa, and J. C. Machado, "A distributed approach for privacy preservation in the publication of trajectory data categories and subject descriptors," pp. 1–8, 2015.

[142] Y. Cao, M. Yoshikawa, Y. Xiao, and L. Xiong, "Quantifying differential privacy in continuous data release under temporal correlations," IEEE Transactions on Knowledge and Data Engineering, vol. 31, pp. 1281–1296, 2019.

[143] N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, "Optimal geoindistinguishable mechanisms for location privacy," Proceedings of the ACM Conference on Computer and Communications Security, pp. 251–262, 2014.

[144] K. Dong, T. Guo, H. Ye, X. Li, and Z. Ling, "On the limitations of existing notions of location privacy," Future Generation Computer Systems, vol. 86, pp. 1513–1522, 2018.

[145] Y. Li, N. Vishwamitra, B. P. Knijnenburg, H. Hu, and K. Caine, "Blur vs. block: Investigating the effectiveness of privacy-enhancing obfuscation for images,"

IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, vol. 2017-July, pp. 1343–1351, 2017.

[146] H. For, T. H. E. Degree, and I. Of, "Data privacy for big automotive data," 2017.

[147] C. Huang, P. Kairouz, X. Chen, L. Sankar, and R. Rajagopal, "Context-aware generative adversarial privacy," Entropy, vol. 19, 12 2017.

[148] K. Emura and T. Hayashi, "Road-to-vehicle communications with timedependent anonymity: A lightweight construction and its experimental results," IEEE Transactions on Vehicular Technology, vol. 67, pp. 1582–1597, 2018.

[149] D. Huang, S. Misra, M. Verma, and G. Xue, "Pacp: An efficient pseudonymous authentication-based conditional privacy protocol for vanets," IEEE Transactions on Intelligent Transportation Systems, vol. 12, pp. 736–746, 2011.

[150] M. Burmester, "Strengthening privacy protection in vanets," pp. 508–513, 2008.

[151] S. Jiang, Y. Chen, J. Yang, C. Zhang, and T. Zhao, "Mixture variational autoencoders," Pattern Recognition Letters, vol. 128, pp. 263–269, 12 2019.

[152] G. Kamath, O. Sheffet, V. Singhal, and J. Ullman, "Differentially private algorithms for learning mixtures of separated gaussians," 2020 Information Theory and Applications Workshop, ITA 2020, 2020.

[153] F. Liu, C. Mathematics, and N. Dame, "Model-based differentially private data synthesis," vol. 46556, 2014.

[154] K. Nissim, S. Raskhodnikova, and A. Smith, "Smooth sensitivity and sampling in private data analysis," Proceedings of the Annual ACM Symposium on Theory of Computing, pp. 75–84, 2007.

[155] J. Hajny, L. Malina, and P. Dzurenda, "Practical privacy-enhancing technologies," pp. 60–64, 2015.

[156] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "Netvlad: Cnn architecture for weakly supervised place recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 40, pp. 1437–1451, 6 2018.

[157] S. RotaBulo and I. M. Bomze, "Infection and immunization: A new class of evolutionary game dynamics," p. 193–211, 2011.

[158] G. Goos, J. Hartmanis, J. Van, L. E. Board, D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, E. Zurich, J. C. Mitchell, M. Naor, O. Nierstrasz, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, and G. Weikum, "Lncs 3757 - energy minimization methods in computer vision and pattern recognition," Lecture Notes in Computer Science, 2009, Volume 5681, ISBN : 978-3-642-03640-8.

[159] R. Raja, S. Kumar, and M. R. Mahmood, "Color object detection based image retrieval using roi segmentation with multi-feature method," Wireless Personal Communications, vol. 112, pp. 169–192, 5 2020.

[160] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope *," International Journal of Computer Vision, vol. 42, pp. 145–175, 2001.

[161] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," IEEE Transactions on Image Processing, vol. 13, pp. 600–612, 4 2004.

[162] S. Vempala and G. Wang, "A spectral algorithm for learning mixtures of distributions," pp. 113–122, 2002.

[163] K. Nissim, U. Stemmer, and S. Vadhan, "Locating a small cluster privately," vol. 26-June-01-July-2016, pp. 413–427, Association for Computing Machinery, 6 2016.

[164] G. Kamath, J. Li, V. Singhal, and J. Ullman, "Privately learning highdimensional distributions," 2019.

[165] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," vol. 2016-December, pp. 3213–3223, IEEE Computer Society, 12 2016.

[166] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," 11 2016.

DeepClean Implementation Code

```
#Use DPGMM Model to generate Private Image Segmentation Samples
import numpy as np
import cv2
import DPGMM as dpGmm

image_path = r'path_to_test_dataset'
img = cv2.imread(image_path)
re_img = img.reshape((-1, 3))
dpGmm_model = dpGmm(ncomps=10, comps_name=comps, priv_budget=0.01,
max_iter=100).fit(re_img)
dpGmm_labels = dpGmm.predict(re_img)

#Convert the flatten image to its original image to
#reconstruct dp segmented image

output_img = dpGmm_labels.reshape(img.shape[0],
img.shape[1])
cv2.imwrite(segmented_img.jpg, output_img)


#DPGMM Class
```

```
# Import the libraries
import numpy as np
import numpy.linalg as nl
import numpy.array_split as nas
class GMM:
    #Implement Private Gaussian Mixture Model
    def __init__(self, ncomps, max_iter = 100, comp_names=None,
    priv_budget):
        #Initialise the model and set parameters
        self.ncomps = ncomps
        self.max_iter = max_iter
        self.priv_budget = priv_budget
        if comp_names == None:
            self.comp_names = [f"comp{index}" for index
            in range(self.ncomps)]
        else:
            self.comp_names = comp_names
        # The pi list holds the fraction of the dataset
        #for every cluster
        self.pi = [1/self.ncomps for comp in range(self.ncomps)]

    def multivariate_normal(self, D, mean_vector,
```

```
covariance_matrix):
    #Implement multivariate normal for the row
    #vector, mean vector and covariance matrix
    return (2*np.pi)**(-len(D)/2)*nl.det(covariance_
    matrix)**(-1/2)*np.exp(-np.dot(np.dot((D-mean_ve
    ctor).T, nl.inv(covariance_matrix)),
    (D-mean_vector))/2)

def fit(self, D):
    # Train the model by passing the 2-d numpy array #data
    # Split data in ncomps subsets
    new_D = nas(D, self.ncomps)
    # Compute the initial mean-vector and covarience #matrix
    self.mean_vector = [np.mean(d, axis=0) for d in new_D]
    self.covariance_matrixes = [np.cov(d.T) for d in new_D]
    # Delete the new_D matrix
    del new_D
    for iter in range(self.max_iter):
        # Perform E-STEP
        # Initialise the r matrix, where every row #contains the
        # for every cluster for this row
        self.r = np.zeros((len(D), self.ncomps))
```

```
# Calculate the r matrix
for n in range(len(D)):
    for k in range(self.ncomps):
        self.r[n][k] = self.pi[k] *
        self.multivariate_normal(D[n],
        self.mean_vector[k],
        self.covariance_matrixes[k])
        self.r[n][k] /=
        sum([self.pi[j]*self.multivariate_no
        rmal(D[n], self.mean_vector[j],
        self.covariance_matrixes[j]) for j in
        range(self.ncomps)])
# Calculate the N
N = np.sum(self.r, axis=0)



# Perform E–STEP
# Initializing the mean vector as a zero #vector
self.mean_vector = np.zeros((self.ncomps, len(D[0])))
# Update the mean vector
for k in range(self.ncomps):
    for n in range(len(D)):
```

```
                self.mean_vector[k] += self.r[n][k] * D[n]
        self.mean_vector =
        [1/N[k]*self.mean_vector[k] for k in
        range(self.ncomps)]
        # Initialise the covariance matrix list
        self.covariance_matrixes =
        [np.zeros((len(D[0]), len(D[0]))) for k in
        range(self.ncomps)]
        # Update the covariance matrices
        for k in range(self.ncomps):
            self.covariance_matrixes[k] = np.cov(D.T,
            aweights=(self.r[:, k]), ddof=0)

        self.covariance_matrixes =
        [1/N[k]*self.covariance_matrixes[k]
        for k in range(self.ncomps)]
        # Update the pi list
        self.pi = [N[k]/len(D) for k in range(self.n_comps)]



def predict(self, D):
```

```
        # Predict clusters using the 2-d numpy array data
        probabilities = []
        for n in range(len(D)):
            probabilities.append([self.multivariate_norm
            al(D[n], self.mean_vector[k],
            self.covariance_matrixes[k])

                                for k in range(self.ncomps)])
        cluster = []
        for proba in probabilities:
            # Check for private clusters and call the
            #noise adding method
            if proba.index is private
                addGaussainNoise(probas)
            cluster.append(self.comp_names[proba.index(max(proba))])
        return cluster



#Encoder and Decoder model design
import numpy as np
import tensorflow as tf
```

```
from tensorflow.keras.initializers import he_normal
from tensorflow.keras.layers import BatchNormalization,
Conv2D, Conv2DTranspose, Flatten

import os
import pathlib
import time
import datetime
from matplotlib import pyplot as plt
from IPython import display
from tensorflow.keras import layers, losses

class Encoder(tf.keras.Model):

  def __init__(self, latent_dim):

    super(Encoder, self).__init__()

    self.enc_block_1 = Conv2D(

                              filters=64,
```

```
                kernel_size=4,

                strides=(2, 2),

                padding = 'same',

                kernel_initializer=he_normal())

self.enc_block_2 = Conv2D(

                filters=128,

                kernel_size=4,

                strides=(2, 2),

                padding = 'same',

                kernel_initializer=he_normal())

self.enc_block_3 = Conv2D(
```

```
                filters=256,
                kernel_size=4,
                strides=(2, 2),
                padding = 'same',
                kernel_initializer=he_normal())
        self.enc_block_4 = Conv2D(
                filters=512,
                kernel_size=4,
                strides=(2, 2),
                padding = 'same',
                kernel_initializer=he_normal())
```

```
self.enc_block_5 = Conv2D(
                filters=512,
                kernel_size=4,
                strides=(2, 2),
                padding = 'same',
                kernel_initializer=he_normal())
self.enc_block_6 = Conv2D(
                filters=512,
                kernel_size=4,
                strides=(2, 2),
                padding = 'same',
```

```
                kernel_initializer=he_normal())


    self.flatten = tf.keras.layers.Flatten()

    self.dense = tf.keras.layers.Dense(latent_dim + latent_dim)

def __call__(self, conditional_input, latent_dim, is_train):

    # Encoder block 1

    x = self.enc_block_1(conditional_input)

    x = BatchNormalization(trainable = is_train)(x)

    x = tf.nn.leaky_relu(x)

    # Encoder block 2

    x = self.enc_block_2(x)
```

```
x = BatchNormalization(trainable = is_train)(x)

x = tf.nn.leaky_relu(x)

# Encoder block 3

x = self.enc_block_3(x)

x = BatchNormalization(trainable = is_train)(x)

x = tf.nn.leaky_relu(x)

# Encoder block 4

x = self.enc_block_4(x)

x = BatchNormalization(trainable = is_train)(x)

x = tf.nn.leaky_relu(x)

# Encoder block 5
```

```
x = self.enc_block_5(x)

x = BatchNormalization(trainable = is_train)(x)

x = tf.nn.leaky_relu(x)

# Encoder block 6

x = self.enc_block_6(x)

x = BatchNormalization(trainable = is_train)(x)

x = tf.nn.leaky_relu(x)




x = self.dense(self.flatten(x))

return x
```

```
class Decoder(tf.keras.Model):

  def __init__(self):

    super(Decoder, self).__init__()

    #self.batch_size = batch_size

    self.dense = tf.keras.layers.Dense(4*4*1)

    self.reshape = tf.keras.layers.Reshape(target_shape=(4, 4, 1))

    self.dec_block_1 = Conv2DTranspose(

            filters=512,

            kernel_size=4,

            strides=(2, 2),

            padding='same',
```

```
            kernel_initializer=he_normal())

        self.dec_block_2 = Conv2DTranspose(

            filters=512,

            kernel_size=4,

            strides=(2, 2),

            padding='same',

            kernel_initializer=he_normal())

        self.dec_block_3 = Conv2DTranspose(

            filters=512,

            kernel_size=4,

            strides=(2, 2),
```

```
            padding='same',

            kernel_initializer=he_normal())

self.dec_block_4 = Conv2DTranspose(

            filters=256,

            kernel_size=4,

            strides=(2, 2),

            padding='same',

            kernel_initializer=he_normal())

self.dec_block_5 = Conv2DTranspose(

            filters=128,

            kernel_size=4,
```

```
            strides=(2, 2),

            padding='same',

            kernel_initializer=he_normal())

    self.dec_block_6 = Conv2DTranspose(

            filters=64,

            kernel_size=4,

            strides=(2, 2),

            padding='same',

            kernel_initializer=he_normal())

    self.dec_block_7 = Conv2DTranspose(

            filters=3,
```

```
            kernel_size=4,

            strides=(1, 1),

            padding='same',

            kernel_initializer=he_normal())

    def __call__(self, z_cond, is_train):

      # Reshape input

      x = self.dense(z_cond)

      x = tf.nn.leaky_relu(x)

      x = self.reshape(x)

      # Decoder block 1

      x = self.dec_block_1(x)
```

```
x = BatchNormalization(trainable = is_train)(x)

x = tf.nn.leaky_relu(x)

# Decoder block 2

x = self.dec_block_2(x)

x = BatchNormalization(trainable = is_train)(x)

x = tf.nn.leaky_relu(x)

# Decoder block 3

x = self.dec_block_3(x)

x = BatchNormalization(trainable = is_train)(x)

x = tf.nn.leaky_relu(x)

# Decoder block 4
```

```
x = self.dec_block_4(x)

x = BatchNormalization(trainable = is_train)(x)

x = tf.nn.leaky_relu(x)

# Decoder block 5

x = self.dec_block_5(x)

x = BatchNormalization(trainable = is_train)(x)

x = tf.nn.leaky_relu(x)

# Decoder block 6

x = self.dec_block_6(x)

x = BatchNormalization(trainable = is_train)(x)

x = tf.nn.leaky_relu(x)
```

```
        return self.dec_block_7(x)


class DeepClean(tf.keras.Model):

    def __init__(self,

        encoder,

        decoder,

        latent_dim,

        label_dim = [256, 256, 3],

        batch_size = 1,

        beta = 1,

        image_dim = [256, 256, 3]):

        super(DeepClean, self).__init__()
```

```
        self.encoder = encoder

        self.decoder = decoder

        self.label_dim = label_dim

        self.latent_dim = latent_dim

        self.batch_size = batch_size

        self.beta = beta = 1

        self.image_dim = image_dim = [256, 256, 3]
    def __call__(self, inputs, is_train):

        input_img, input_label, conditional_input =
        self.conditional_input(inputs)

        z_mean, z_log_var =
        tf.split(self.encoder(conditional_input,
        self.latent_dim, is_train), num_or_size_splits=2,
```

```
axis=1)

z_cond = self.reparametrization(z_mean,
z_log_var, input_label)

logits = self.decoder(z_cond, is_train)


recon_img = tf.nn.sigmoid(logits)


# Compute Loss#

latent_loss = -0.5 * tf.reduce_sum(1 + z_log_var
- tf.square(z_mean) - tf.exp(z_log_var), axis=-1)
# KL divergence

reconstr_loss = np.prod((64,64)) *
tf.keras.losses.binary_crossentropy(tf.keras.bac
kend.flatten(input_img),
tf.keras.backend.flatten(recon_img)) # over
weighted MSE
```

```
loss = reconstr_loss + self.beta * latent_loss
# weighted ELBO loss

loss = tf.reduce_mean(loss)


return {
            'recon_img': recon_img,
            'latent_loss': latent_loss,
            'reconstr_loss': reconstr_loss,
            'loss': loss,
            'z_mean': z_mean,
            'z_log_var': z_log_var
        }
```

```
def conditional_input(self, inputs):

    """ Builds the conditional input and returns the
    original input images, their labels and the
    conditional input from the DPGMM model."""

    input_img =
    tf.keras.layers.InputLayer(input_shape=self.image_dim,
    dtype = 'float32')(inputs[0])

    input_label =
    tf.keras.layers.InputLayer(input_shape=self.label_dim,
    dtype = 'float32')(inputs[1])

    input_label = Flatten()(inputs[1])

    labels = tf.reshape(input_label, [-1, 1, 1])
    #batch_size, 1, 1, label_size

    ones = tf.ones([inputs[0].shape[0]] +
```

```
        self.image_dim[0:-1]) #batch_size, 128, 128,
        label_size

        labels = ones * labels #batch_size, 128, 128,
        label_size

        conditional_input = tf.keras.layers.InputLayer(input_shape=
        (self.ima
        ge_dim[0], self.image_dim[1], self.image_dim[2]),
        dtype = 'float32')(tf.concat([inputs[0], labels],
        axis=3))



        return input_img, input_label, conditional_input


def reparametrization(self, z_mean, z_log_var,
input_label):

        """ Performs the riparametrization trick"""
```

```
        eps = tf.random.normal(shape =
        (input_label.shape[0], self.latent_dim), mean =
        0.0, stddev = 1.0)

        z = z_mean + tf.math.exp(z_log_var * .5) * eps

        z_cond = tf.concat([z, input_label], axis=1) #
        (batch_size, label_dim + latent_dim)

        return z_cond



#DeeClean Model usage
from tensorflow import keras
encoder = Encoder(128)
decoder = Decpder()
model = DeepClean( encoder, decoder,
        label_dim = [256, 256, 3],
        image_dim = [256, 256, 3],
```

```
        latent_dim = 128,
        beta = 0.65)
optimizer = keras.optimizers.Adam(learning_rate=0.01)
```