Privacy-Preserving Visual Analysis: Training Video Obfuscation Models Without Sensitive Labels

Sander De Coninck[1*], Wei-Cheng Wang[1], Sam Leroux[1], Pieter Simoens[1]

[1*]IDLab, Department of Information Technology, Ghent University - imec,
Technologiepark 126, Ghent, B-9052, Belgium.

*Corresponding author(s). E-mail(s): sander.deconinck@ugent.be;
Contributing authors: weicheng.wang@ugent.be; sam.leroux@ugent.be;
pieter.simoens@ugent.be;

Abstract

Visual analysis tasks, including crowd management, often require resource-intensive machine learning models, posing challenges for deployment on edge hardware. Consequently, cloud computing emerges as a prevalent solution. To address privacy concerns associated with offloading video data to remote cloud platforms, we present a novel approach using adversarial training to develop a lightweight obfuscator neural network. Our method focuses on pedestrian detection as an example of visual analysis, allowing the transformation of video frames on the camera itself to retain only essential information for pedestrian detection while preserving privacy. Importantly, the obfuscated data remains compatible with publicly available object detectors, requiring no modifications or significant loss in accuracy. Additionally, our technique overcomes the common limitation of relying on labeled sensitive attributes for privacy preservation. By demonstrating the inability of pedestrian attribute recognition models to detect attributes in obfuscated videos, we validate the efficacy of our privacy protection method. Our results suggest that this scalable approach holds promise for enabling camera usage in video analytics while upholding personal privacy.

Keywords: Privacy-Preserving Edge Computing, Cloud-Edge Collaboration, Visual Analysis, Pedestrian detection, Pedestrian Attribute Recognition

1 Introduction

Privacy has become a critical concern in the era of smart cities and ubiquitous video surveillance systems. As cities become more connected and data-driven, individuals' privacy rights must be safeguarded, especially regarding their movements and activities in public and semi-public spaces. Many applications, such as crowd management or people counting, rely on video data for processing. However, this video data contains sensitive privacy aspects that could be exploited for malicious purposes beyond the original task, such as person identification or racial profiling [1, 2]. The growing usage of AI for smart city applications, coupled with the prevalence of cloud computing, further exacerbates these privacy concerns [3].

There are two potential solutions to address privacy and security concerns in camera analytics. One approach is to conduct all computations on the camera device and only transmit the results, which can prevent data leakage. However, this approach has limitations as edge devices have fewer computational resources and are more prone to wear and tear. Providing all cameras with embedded processors capable of running analytics is, therefore, a costly solution. Moreover, to safeguard the data from being compromised during computation, it is necessary to work within a Trusted Execution Environment (TEE), which has severe memory limitations [4].

An alternative solution is to integrate a privacy-preserving technique on the edge device, which is less computationally intensive and more suitable for TEEs while offloading the majority of the computation to a cloud model. This approach leverages the advantages of cloud computing, such as lower computational costs, better maintainability, and scalability, while requiring an additional component to provide privacy protection. The framework proposed in this paper belongs to this category of solutions, which we apply to the visual analysis task of pedestrian detection.

In this category, researchers have used anonymization, encryption, and aggregation techniques on surveillance footage to protect individuals' privacy while extracting valuable insights from the data [5–8]. However, the models used in this category of works are often explicitly trained to prohibit inferring specific sensitive attributes, which has several drawbacks. Firstly, this necessitates defining all sensitive attributes beforehand, making it vulnerable to oversight. Secondly, these techniques need machine learning models to infer these private attributes, and, subsequently, labelled datasets of all private attributes which requires compliance with data protection legislation. Finally, existing strategies frequently require modifications to the machine learning model of the allowed task, making them incompatible with existing architectures for these tasks and more costly to develop.

This paper proposes a novel approach for privacy-preserving edge-cloud visual analysis based on Generative Adversarial Privacy (GAP) [9]. We apply our technique to pedestrian detection, as this is an indicative task for smart city applications involving human data. Note that the framework can be applied to any task. Our method is designed to maintain high detection accuracy while minimizing the amount of personal data collected and processed, making it suitable for use in smart cities and other urban applications. Critically, our privacy-preserving scheme does not require access to sensitive task labels and focuses solely on allowing the task (e.g. pedestrian detection) in an opt-in manner by preventing adversaries from recreating the original data from the privatized form. Furthermore, our technique does not require modification of the downstream model, making it compatible with existing utility models and allowing the usage of third-party software.

We validate privacy preservation by training Pedestrian Attribute Recognition (PAR) models on obfuscated data using labels from the original data, positing that they will be unable to learn on filtered data since most attributes are related to identification but are not necessary for detection of the presence of *a* pedestrian. PAR accuracy was chosen as an indication of privacy as the attributes recognized can be considered sensitive, such as gender, age, etc. Moreover, PAR is regarded as a foundational computer vision block in intrusive tasks such as person re-identification and tracking [10]. Figure 1 showcases the intended use of our work.

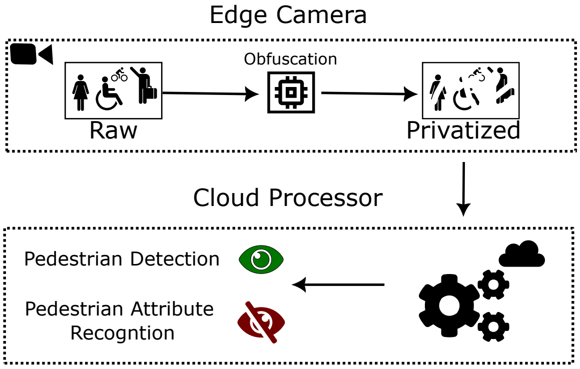

Fig. 1 Intended deployment of our obfuscation technique. The camera footage is obfuscated to a privatized form, which allows a certain *downstream utility* (e.g. pedestrian detection) but disallows other, possibly malicious, tasks to be performed (e.g. inferring pedestrian attributes)

We compare our technique against both classic obfuscation methods like blurring, noising, quantizing and pixelating as well as deep-learning-based techniques focusing on pedestrian anonymization. Subsequently, we verify additional wanted properties of our technique when used in an edge-cloud setup, such as efficiency in comparison to pedestrian detection models and generalizability across multiple object detection models and cameras. Lastly, we investigate the effect of model complexity on the privacy-utility tradeoff.

The main contributions of our work are as follows:

- Our adversarial obfuscation allows pedestrian detection with minimal accuracy degradation while significantly decreasing the information on pedestrian attributes in the video transmitted to the cloud-based model.
- Our method provides significantly better privacy protection than classic obfuscation techniques, given the same utility. The adversarial obfuscator outperforms the deep-learning based methods while requiring less time to execute and memory to store.
- Our method can generalize over multiple cameras from the same dataset. It is, however, not compatible with different person detection models other than the one it was trained for.
- The complexity of the obfuscator model can be used to tune the privacy-accuracy tradeoff slightly.

The outline of this paper is as follows: Section 2 describes the related works in privacy-preserving machine learning and pedestrian detection. Subsequently, we describe the architecture of our privacy-preserving scheme and evaluation technique in Section 3 and 3.2. The experiments and their results are described in Section 4. Finally, we conclude our paper and look to future works in Section 5.

2 Related works

2.1 Privacy-Preserving Machine Learning

Apart from pedestrian detection, privacy-preserving machine learning techniques are available for several applications. The main techniques involve differential privacy [11], homomorphic encryption [12], secure multi-party computation [13], or reliance on information-theoretic properties [14].

As our work falls in the latter group, we will highlight some works in this section. In the context of video data, a common approach is to use a two-step process, where sensitive regions (e.g. faces) are first detected and then modified using inpainting techniques [15–17] or more basic methods such as blurring or pixelization. The former approach's use of two deep neural networks (DNNs) makes it challenging for real-time processing, while the latter approach is vulnerable to deep learning attacks [18]. An alternative one-step solution is generative adversarial privacy [9]. GAP involves training a privatizing network to degrade the data and an adversary network that attempts to infer the sensitive data from the degraded version. However, its usage is limited by the need to define and label all sensitive aspects. Recently some works have managed to circumvent these issues when working with audio data [19], or for image classification [20]. Though, this solution has yet to be applied to the domain of object detection on video data.

2.2 Privacy-Preserving Pedestrian Detection

Privacy-preserving pedestrian detection has become an increasing concern in developing surveillance technology. Several researchers have explored innovative techniques to balance the need for accurate detection while protecting individual privacy. Yuan et al. [5] utilized differential privacy to address privacy concerns in pedestrian detection by adding Gaussian noise to the entire frame. Other works utilize a two-step approach where people are first detected and subsequently replaced with an anonymized version [7, 8]. Chan et al. [21] proposed a privacy-preserving approach to crowd monitoring that did not require person detection but required special-purpose cameras that output low-level features. Kieu et al. [22] suggested using thermal cameras to protect privacy, but did not substantiate their claim that person identification is difficult or impossible. Bentafat et al. [6] provide a solution for real-time privacy-preserving video surveillance by encrypting the regions of faces, though they do not address any other attributes that can be used for recognition. Lastly, Yang et al. [23] utilize homomorphic encryption, to encrypt images while allowing the extraction of Histogram of Oriented Gradients features, which can then be used by an SVM model to detect pedestrians.

3 Architecture

3.1 Model architecture

The goal of our technique is to transform frames of a video in a manner that enables the detection of pedestrians while simultaneously removing any superfluous information. Critically, we aim to achieve this without modifying the pedestrian detection model, making it compatible with third-party models, and without relying on a dataset of sensitive attributes. To this end, we have applied adversarial training to obtain a filtered version of our input frames, which is achieved through the use of an autoencoder, referred to as the "obfuscator", denoted as O. This autoencoder transforms the frame in a manner that permits pedestrian detection while simultaneously hindering the reconstruction of the original image using a second autoencoder, referred to as the "deobfuscator" and denoted as D. By training both autoencoders in tandem, we posit that the obfuscator learns a transformation that solely retains the required features for pedestrian detection. Figure 2 showcases the architecture of our proposed approach, which we term the adversarial obfuscator.

The deobfuscator reconstructs the original image from its obfuscated version, essentially acting as an adaptable mutual information estimator between the two. We experimented with different image similarity metrics, including SSIM [24], LPIPS [25], DISTS [26], VIF [27], and Mean Squared Error (MSE), to train the deobfuscator. However, we observed minimal differences in the results and thus chose the simplest metric. Therefore, the deobfuscator is trained by minimizing the pixel-wise MSE between the original image X and the deobfuscated image. The loss function, denoted by L_{rec}, is defined as follows:

$$L_{rec} = \frac{1}{n} \sum_{i=1}^{n} (X_i - D(O(X_i)))^2.$$

The obfuscator is trained by minimizing the loss of the object detection model L_{obj} and maximizing the reconstruction loss of the deobfuscator. The loss function is defined as follows, with α_{obj} and α_{rec} as weight factors:

$$L_O = \alpha_{obj} L_{obj} - \alpha_{rec} L_{rec}.$$

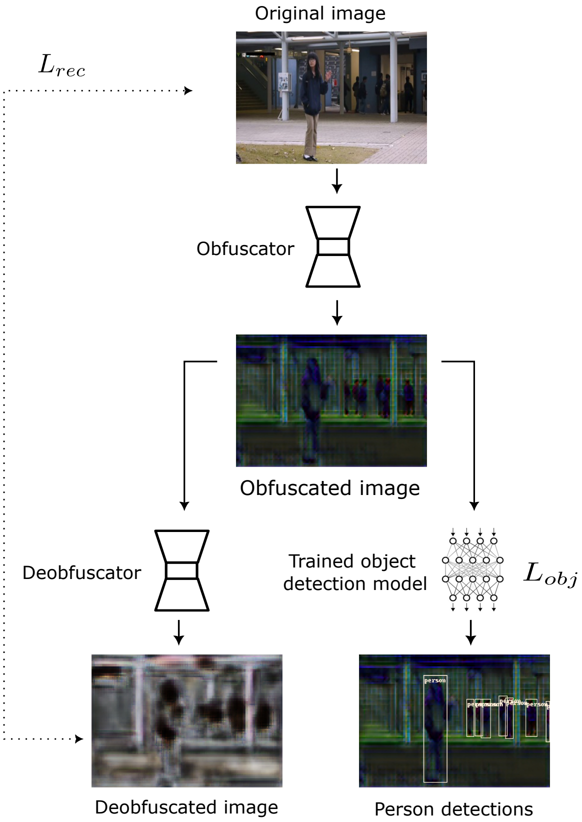

Fig. 2 The architecture of the adversarial obfuscator. The obfuscator transforms the input image to allow pedestrian detection but disallow an adversary (the deobfuscator) to reconstruct the original image using its obfuscated image.

Our training procedure thus requires access to the weights of the object detection model for performing gradient descent, and to the loss function on which this model was trained to calculate L_O. These requirements are an inherent restriction of the GAP framework. The obfuscator and deobfuscator are constructed based on the MobileNet [28] architecture, which was designed for devices with limited computational resources making the obfuscation process more suited for edge devices and inference in a TEE.

3.2 Evaluation

The evaluation of our technique is twofold. We want to evaluate the intended task (utility), i.e. the accuracy of detecting pedestrians on obfuscated images. Meanwhile, the evaluation of the privacy protection has to be considered. As both evaluations require labelled data, we created our own dataset from publicly available CCTV-like videos and created pseudo-labels for the target task and the sensitive labels using pre-trained pedestrian detection and attribute recognition models.

3.2.1 Utility

The utility is defined by the performance of the intended task, in this case, pedestrian detection. We trained the obfuscator to be compatible with a pre-trained object detection model. Evaluation of the utility can thus be achieved using the detection accuracy of this pre-trained neural network on the obfuscated frames. We measured the performance of this network by calculating the Average Precision (AP) for the pedestrian class. We did this using the implementation of Cartucho et al. [29], which uses the PASCAL VOC criterium.

3.2.2 Privacy

Obtaining a metric to evaluate the performance of privacy protection is challenging due to the absence of publicly available datasets containing ground truth labels of privacy-sensitive attributes, as well as a lack of a precise definition of what constitutes such attributes. To address this issue, we assess the effectiveness of privacy protection by evaluating the ability of pedestrian attribute recognition (PAR) models to learn from datasets with obfuscated images. PAR models are designed to predict various attributes of individuals; e.g. age, gender, clothing, etc.; and are often used as a foundation for privacy-invading tasks, including person re-identification [10].

The performance of the PAR models is measured using two groups of metrics: instance-level and attribute-level. Instance-level metrics are used to assess the model's ability to classify the attributes of individual persons in the dataset. Attribute-level metrics evaluate the model's effectiveness in classifying the attributes themselves over the entire dataset. The instance-level attributes include Accuracy (Acc), Precision (Prec), Recall (Rec) and F1. Attribute-level consists of the mean Accuracy (mA), which can be calculated as follows, with M being the number of attributes:

$$mA = \frac{1}{M}\sum_{j=1}^{M}\frac{1}{2}(\frac{TP^j}{TP^j + FN^j} + \frac{TN^j}{TN^j + FP^j}).$$

It is crucial to consider the ideal values corresponding to the metrics used to evaluate the filtering technique, as they may not be immediately intuitive. Although lower values may appear to indicate better filtering, it is important to note that the labels for most sensitive attributes in PAR are heavily unbalanced. Hence, an adversary with prior knowledge could achieve better-than-random accuracy by consistently guessing the majority class. Precision and recall metrics measure the fraction of correct positive predictions and the number of positive instances retrieved, respectively. However, in datasets with a large proportion of negative samples, these metrics may not have enough samples to calculate accurately. The mean accuracy metric is a better measure in this case, as it takes the average between the positive and negative recall, requiring both positive and negative samples to be well classified. Therefore, we focus on the mA metric in this paper, where a value of 0.5 is considered optimal for filtered data.

4 Experiments

We utilized three datasets for experimentation: the Avenues dataset [30], ShanghaiTech Campus [31] dataset and WILDTRACK [32] dataset. These datasets comprise CCTV camera footage featuring a high volume of pedestrian activity, where pedestrians are close enough to the camera for their attributes to be recognized. For the Avenues dataset,which contains 16 training videos and 21 testing videos, we used 8 videos of the test set as a validation set. As for the ShanghaiTech and WILDTRACK datasets, we divided the data of every camera into train/validation/test sets with a ratio of 80/10/10, ensuring videos were not split over different sets. The WILDTRACK dataset contains data of 7 cameras, each with over 30 minutes each of footage. Whereas the ShanghaiTech Campus dataset consists of footage of 11 cameras, of which we selected the 5 cameras that have the most footage available (i.e. more than 20 videos), as others had too little to train and evaluate our technique. Given that these datasets were not originally intended for pedestrian detection, they do not possess any ground-truth labels[1]. As a solution, we employed a pre-trained Faster R-CNN [33] model to obtain pseudo-labels. We retained only those labels with a confidence score of at least 90% to mitigate the presence of overly noisy labels. We resized the videos to 200 × 320 pixels to decrease the computational load and extracted frames at 5 fps.

As for the training details: we alternated between training our obfuscator and deobfuscator for a total of 30 epochs, starting with a learning rate of 1×10^{-3} and decreasing it by a factor of 10 at the halfway point. The AdamW [34] optimizer was used with a weight decay of 1×10^{-6} and PyTorch's automatic mixed precision training. We used 1 as value for both α_{rec} and α_{obj}. PyTorch's pre-trained Faster R-CNN model was used as our object detection model to train the obfuscator. After training, the deobfuscator is discarded, and the obfuscator is used to filter private information in the camera frames. Some example images of obfuscated frames from the three datasets can be seen in Figure 3. Variations in the visual appearance of obfuscations may occur due to the stochastic initialization of both the obfuscator and deobfuscator models.

4.1 Pedestrian detection ability

We use the AP metric, to assess the opt-in effectiveness of our framework. The results are shown in Table 1. It is important to note that, since our ground-truth labels were acquired through an object detection model, the reported accuracies should be interpreted with caution. Any persons not detected in the original data may be detected in the obfuscated data, but will still be considered inaccurate predictions. The AP values for person detections range mostly around 90%, indicating that most pedestrians are detected the same as if the images were not obfuscated. The WILDTRACK cameras exhibit the highest degree of variability in accuracy across cameras. This can be attributed to the fact that this is the dataset with the highest density of pedestrians visible.

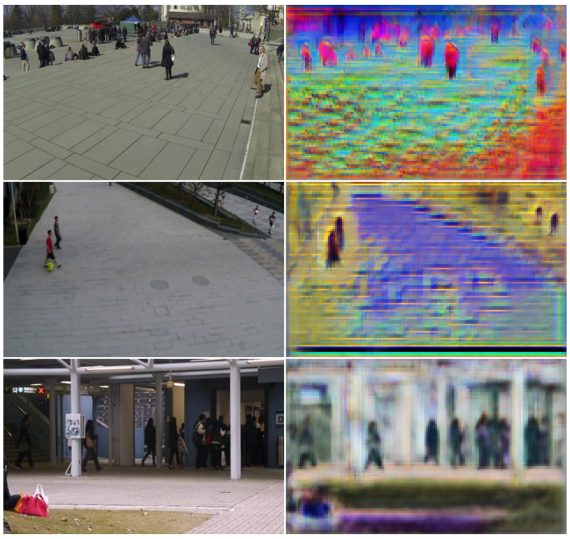

Fig. 3 Original (left) and obfuscated (right) frames of the WILDTRACK, ShanghaiTech Campus and Avenues datasets. The obfuscation appearance can vary across datasets due to the stochastic initialization of the obfuscator and deobfuscator models.

Table 1 Relative pedestrian detection accuracy on obfuscated frames. The obfuscation brings forth a small reduction in pedestrian detection accuracy.

Dataset	Camera	Person AP
Avenues		89.62
ShanghaiTech Campus	1	89.26
	4	83.86
	5	93.28
	6	91.70
	8	87.72
WILDTRACK	1	83.35
	2	90.45
	3	88.36
	4	71.32
	5	83.63
	6	71.21
	7	80.97

4.2 Assessing privacy through pedestrian attribute recognition

To test PAR performance on obfuscated data, we employ two baseline models, the model of Jia et al. [35], and the Visual Textual Baseline (VTB) [36]. We trained these models on the UPAR [37] dataset and used them on cropped-out persons from our pedestrian detection datasets to obtain ground truth (pseudo-)labels. These models are trained to detect 40 binary attributes, such

[1] The WILDTRACK dataset does contain ground-truth values, however, due to the limited amount available and the fact that only a particular section of pedestrians was annotated, we chose not to use these.

as various age indicators, accessories, clothing etc. We then created a new dataset by replacing the person crops with obfuscated versions, simulating an attack by an adversary that has access to the obfuscation model. Our hypothesis is that if a PAR model is unable to train on the obfuscated dataset, it no longer contains the data necessary for this task, suggesting that the obfuscation successfully removes sensitive data, such as those necessary for attribute recognition. We removed the most unbalanced attributes (i.e. when more than 99% of the samples are entirely in one category), and crops smaller than 20 × 20 pixels, as these prove to be too small to detect any attributes. The results of pedestrian attribute recognition on the original, obfuscated and reconstructed frame for both PAR models are shown in Table 2.

Table 2 Comparison of PAR mA using common PAR models trained on original, obfuscated and reconstructed data. The capacity to learn pedestrian attributes on obfuscated data is significantly reduced.

	Jia et al. [35]			VTB [36]		
	Orig.	Obf.	Rec.	Orig.	Obf.	Rec.
Dataset						
Avenues	76.28	66.51	60.98	80.73	66.40	55.75
ShanghaiTech Campus 1	76.85	57.17	55.96	80.38	56.31	60.98
ShanghaiTech Campus 4	65.83	53.23	52.34	65.06	50.17	55.96
ShanghaiTech Campus 5	78.88	60.48	57.77	78.47	62.03	52.34
ShanghaiTech Campus 6	71.49	59.62	57.49	69.97	57.74	57.77
ShanghaiTech Campus 8	71.19	58.20	56.57	79.00	52.47	57.49
WILDTRACK 1	74.06	56.19	54.34	72.21	53.09	56.57
WILDTRACK 2	79.85	60.59	56.96	82.56	53.50	54.34
WILDTRACK 3	78.57	58.04	57.29	80.55	56.02	56.96
WILDTRACK 4	74.21	54.29	54.67	70.67	51.40	57.29
WILDTRACK 5	75.53	57.58	56.00	76.41	55.79	54.67
WILDTRACK 6	73.78	55.57	54.32	74.93	53.20	56.00
WILDTRACK 7	72.73	57.02	55.75	73.04	53.76	54.32

The results show that there is a significant performance degradation in attribute recognition on obfuscated frames for all datasets, with the mA almost reaching the level of random guessing. This indicates that the obfuscation process removes relevant information for PAR. For the ShanghaiTech and WILDTRACK datasets, the performance drops significantly to around $53 \sim 60\%$ mA for the model of Jia et al. and $50 \sim 56\%$ for VTB, indicating that the model can learn very little from obfuscated frames. The PAR mA on the Avenues dataset is still 66%. Although this is a drop of more than 10%, it is still far from the desired 50%. However, it should be noted that the data in the Avenues dataset is the most unbalanced in terms of attributes, featuring many attributes slightly outside the cutoff for unbalance, which could be one reason why the PAR performance is not close to random.

Detecting pedestrian attributes remains challenging even in the reconstructed frames obtained through deobfuscation. This difficulty stems from the absence of essential information in the input data for the deobfuscator, i.e., the obfuscated frame, due to the data processing inequality. As the deobfuscator cannot re-introduce information, this limitation results in similar or worse pedestrian attribute recognition (PAR). It is important to note that the reconstructed data, despite its origin, does not offer improved suitability for our specific objective, as it significantly impedes effective pedestrian detection.

4.3 Comparison with other obfuscation techniques

When it comes to protecting sensitive information in images, a variety of techniques can be employed to obfuscate the information and make it harder to discern. To ensure a comprehensive and unbiased evaluation, we conducted two sets of experiments. In the first set, we compared our adversarial obfuscator with traditional methods such as blurring, adding noise, quantization, and pixelation. Similar to our approach, these techniques were applied to the entire frame without prior knowledge of the location of sensitive information. However, a key limitation of these traditional methods is that while they may effectively reduce the detection of privacy-sensitive elements, they also compromise utility. To facilitate a meaningful comparison between our technique and these obfuscation methods, we adjusted their parameters to achieve a similar level of utility loss as ours, specifically aiming for a relative pedestrian accuracy of approximately 90%. This entailed using a kernel size of $[9, 9]$ for Gaussian blurring, adding 7.5% noise, quantizing to 8 values, and reducing the image size by one-third for pixelation.

In the second set of experiments, we evaluate our approach against more sophisticated, deep learning-based methods. These methods follow a two-step procedure where individuals are first identified and then anonymized. While these approaches generally offer a better balance between privacy and utility compared to the previous category, they are less adaptable in terms of specifying which privacy-sensitive aspects to safeguard. Additionally, they are more prone to instances where individuals are missed during detection, resulting in their exclusion from anonymization. Moreover, these methods demand substantially higher computational resources, rendering them less suitable for edge deployment. Within this category, we compare our approach with a two-step anonymizer utilizing Mask R-CNN [38] for person segmentation, followed by either blurring or complete removal of individuals through mask replacement. Furthermore, we assess our approach against the current state-of-the-art in realistic whole-body anonymization, represented by DeepPrivacy2 [7].

The results from both experiments conducted on camera 1 of the ShanghaiTech Campus dataset are presented in Table 3. The upper section displays the outcomes of the first experiment set, while those below the dashed line represent the second set. Figure 4 provides a visual comparison of all techniques.

Concerning the classic obfuscation methods, our approach demonstrates the most balanced privacy-utility tradeoff. This is evident in the significantly lower PAR mA compared to other techniques, while maintaining a similar person AP.

Results from the second set of experiments exhibit greater variability. Notably, in the case of Mask R-CNN + blur/mask-out, there is a significant reduction in utility. However, PAR performance remains relatively stable. This may be due to two factors: first, not all individuals are detected and, therefore, not all are anonymized. Second, even after anonymization, the contour of individuals remains discernible, potentially providing the PAR model with valuable cues. For Mask R-CNN, we found that it is mainly the latter case, as almost all persons in the ground truth are detected. DeepPrivacy2 achieves a comparable privacy-utility tradeoff compared to our technique. However, the decline in PAR performance is less pronounced than in our approach. One cause for this is that, for all two-step techniques, the anonymization is only applied to those persons detected. We found that for DeepPrivacy2, over one-third of all persons in the test dataset were not detected, leading to significant privacy leakage. Note that this discrepancy between the Mask R-CNN techniques and DeepPrivacy2, even though DeepPrivacy2 also relies on Mask R-CNN, lies in part to the evaluation. We use data that was labelled using a Faster R-CNN model with the same backbone as the Mask R-CNN (ResNet50), whereas DeepPrivacy2 uses Mask R-CNN with a ResNeXt-101 backbone.

Table 3 Comparison with other privacy-preserving techniques

Technique	Person AP ↑	PAR mA ↓
Original image	100	76.85
Blurring	87.51	73.16
Noise	87.65	74.33
Quantization	87.60	75.34
Pixelisation	91.17	72.60
DeepPrivacy2 [7]	**93.60**	63.60
Mask R-CNN + blur	56.25	69.65
Mask R-CNN + mask-out	27.06	67.72
Adversarial obfuscation (ours)	89.26	**57.17**

Table 4 Execution time on Jetson AGX Orin and number of parameters of deep learning-based obfuscation techniques.

Method	Time (ms)	Params (M)
Adversarial obfuscator (ours)	5.50 ± 1.25	0.04
DeepPrivacy2	613.62 ± 212.82	234.43
Mask R-CNN + mask-out	165.78 ± 19.57	44.18
Mask R-CNN + blur	170.02 ± 22.54	44.18

Finally, it's worth noting that these two-step techniques demand substantially more computational resources compared to ours, which might not always be readily available when working with edge devices. To support this claim, we included the number of parameters of all models (which correspond to memory and storage requirements) and the execution time on the Jetson Orin AGX to anonymize one frame. The findings are presented in Table 4. It's important to highlight that for these techniques, the computation needed depends on the number of pedestrians detected, as the anonymization phase is only triggered for detections. The results show our technique significantly outperforms the alternatives, concerning both execution time as well as memory required to store the model.

4.4 Cross-camera generalization

The present study describes a filtering method specifically tailored to edge device cameras. Given the abundance of such cameras, it is important to assess the ability of the proposed filter to generalize across different cameras, thereby avoiding

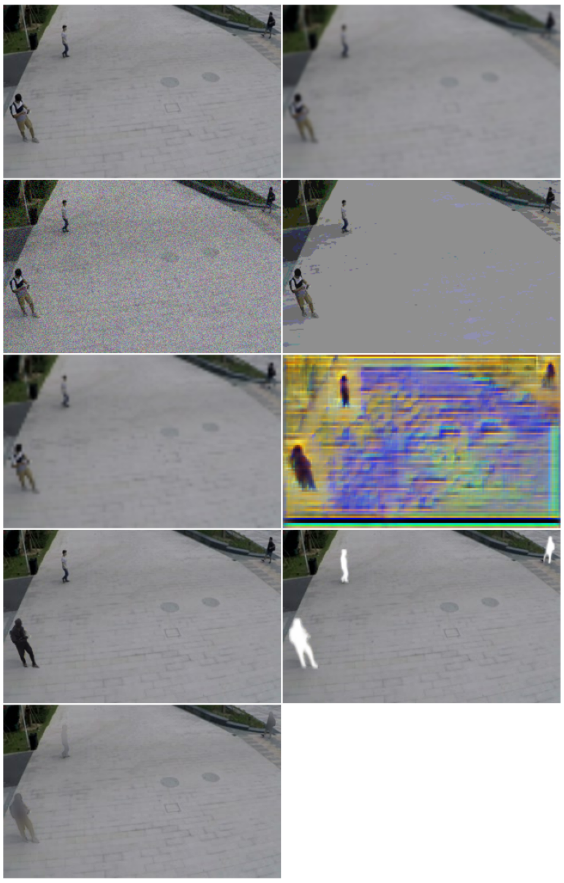

Fig. 4 Obfuscation with other obfuscation techniques. From left to right, top to bottom: Original image, blurred, noised, quantized, pixelized, obfuscated (ours), DeepPrivacy2, Mask-out, Mask + blur

the need for training specific models for each camera. To evaluate cross-camera generalization, two datasets were employed. The first, the ShanghaiTech Campus dataset, consists of footage from 13 cameras placed at various heights and illuminance levels in different locations. To test the model's performance, data from cameras 1-8 were used for training, while data from cameras 9-13 were used for testing. The WILDTRACK dataset offered a different perspective on cross-camera generalization, in which all cameras were situated in different locations but had a view of the same area, resulting in a considerable amount of shared information.

The results of this experiment can be seen in Table 5. We observe generalization on the ShanghaiTech Campus dataset, as the person AP on new cameras is similar to the results in Table 1, where we evaluated on the same cameras. For the WILDTRACK dataset, the results are less conclusive. Camera 7 shows good performance, though for camera 6, there is a more significant decrease in accuracy. This could be attributed to the fact that this camera is directed more towards buildings and less towards the square than the other cameras. These results indicate that to generalize to other cameras, the obfuscator should be trained on a well-diversified set of cameras.

Table 5 Cross-Camera pedestrian detection accuracy. Generalisation is acquired for the ShanghaiTech Campus but not for the WILDTRACK dataset.

Dataset	Train Cameras	Test Camera	Person AP
ShanghaiTech Campus	1-8	9	87.23
		10	83.43
		11	92.53
		12	82.07
		13	87.56
WILDTRACK	1-5	6	60.27
		7	82.04

The results for privacy protection can be seen in Table 6. As the data of the separate ShanghaiTech Campus cameras are too small to both train and test a PAR model, we decided to aggregate all test cameras in one set. The results are similar to those for the single-camera setting, even for the WILDTRACK dataset. We can conclude that our obfuscation method can generalize across cameras, sacrificing slightly in utility but not in privacy protection.

Table 6 Cross-Camera PAR mean accuracy. Even on different cameras, obfuscation removes attribute information.

	Jia et al. [35]		VTB [36]	
Dataset	Orig.	Obf.	Orig.	Obf.
ShanghaiTech Campus 9-13	74.33	57.18	78.23	55.64
WILDTRACK 6	70.76	57.21	73.83	55.16
WILDTRACK 7	70.66	55.70	74.93	54.03

4.5 Suitability for edge processing

Our obfuscation technique is designed to be used on edge devices, while pedestrian detection is performed on the cloud. Therefore, we would expect our obfuscation to be considerably more lightweight than the pedestrian detection model used. Otherwise, one might argue that it would be more privacy-safe to run detection on the edge and send through the results. Note, however, that execution on the cloud brings considerable benefits, such as better maintainability and cheaper compute.

To evaluate the suitability of our technique for edge devices, two metrics of DNNs are examined. The first metric is the number of Multiply Accumulates (MACs), which refers to the number of arithmetic operations that involve multiplication and addition. This metric is a reliable indicator of the computation required to process the data. The second metric is the number of model parameters, which indicates the memory cost of storing the model. The latter metric is particularly critical when considering the implementation of the obfuscator in a TEE. Smaller values for both metrics indicate that a model is better suited for edge deployment. In addition to these metrics, we measure the time to process a single frame on a Jetson Orin AGX device to indicate compatibility on an edge device.

Table 7 shows an overview of these metrics. Values are calculated by the ptflops [39] package. The results show that our proposed obfuscation model is considerably smaller than object detection models. Most notably, our model uses almost 100 times fewer parameters than the Yolov8 nano model, making it highly suitable for deployment on edge devices.We also require significantly less compute, and can execute a single frame in around 5.5ms.

Table 7 Comparions of model complexity with common object detection models. Our is far smaller than the compared models, especially concerning the number of parameters.

Model	MACS (G)	Params (M)	mAP COCO	Time (ms)
Faster R-CNN v1 (Resnet50)	199.79	41.53	37.0	130.48 ± 1.63
Faster R-CNN (MobileNetV3 Large)	8.26	19.33	32.8	49.40 ± 0.65
Faster R-CNN (MobileNetV3 Large 320)	1.16	19.33	22.8	43.71 ± 2.71
FCOS (Resnet50)	205.43	43.71	39.2	113.33 ± 8.07
RetinaNet (ResNet50)	242.66	33.79	36.4	126.04 ± 1.81
SSD (VVG16)	34.92	35.60	25.1	57.03 ± 2.07
YOLOv8n	1.10	3.15	37.3	20.41 ± 2.26
YOLOv8s	11.17	3.70	44.9	18.92 ± 1.83
Adversarial Obfuscator (Ours)	**0.35**	**0.04**	-	5.50 ± 1.25

4.6 Object detection model independence

We train our obfuscator by including the object detection model in the training loop. Nonetheless, the goal is for it to function independently of the detection model. When using a cloud-hosted detection service, the architecture of the object detection model may not be publicly available.

To test the obfuscator's independence, we evaluate its performance on various object detection models with different backbone architectures, after training it on a Faster R-CNN model with a ResNet50 backbone. Table 8 shows the results of this experiment. We see that for most different object detection models, the performance decreases considerably. Only those models with the same backbone and comparable head architectures show some level of compatibility. However, the extent of this compatibility is hard to determine, and it can only be concluded that the obfuscator's generalizability across object detection models is limited.

4.7 Model complexity and the accuracy-privacy tradeoff

In this section, we investigate the feasibility of manipulating the privacy-utility tradeoff by adjusting the complexity of the obfuscator model. Specifically, reducing the model complexity may result in a decreased ability to preserve relevant features, leading to a reduction in person detection accuracy. However, it may also increase privacy protection by decreasing the number of sensitive features that can be retained. The impact of reducing model complexity on utility is depicted in Figure 5, which demonstrates that utility remains relatively stable at around 90% until the models become too small, at which point it declines rapidly. Conversely, Figure 6 demonstrates that reducing the model size has an inverse effect on privacy, as the mA on obfuscated frames remains stable until a certain point, after which it decreases further (thus enhancing privacy).

The aforementioned figures suggest that adjusting the complexity of the obfuscator model can lead to a tradeoff between privacy and utility. However, it is worth noting that the reduction in person detection accuracy ability is considerably less pronounced than the decrease in utility.

Table 8 Cross-model pedestrian detection accuracy on Faster-RCNN guided images. The obfuscator's generalizability across object detection models is limited.

Dataset	Faster R-CNN ResNet50	Faster R-CNN v2 ResNet50	RetinaNet ResNet50	Mask R-CNN ResNet50	Faster R-CNN MobileNetV3 large	Faster R-CNN MobileNetV3 large 320	YOLOv8 n
Avenues	89.62	92.82	5.04	92.72	66.49	62.12	19.79
ShanghaiTech Campus Camera 1	89.18	82.13	11.36	88.09	33.75	45.03	24.27
WILDTRACK Camera 1	82.31	64.42	4.06	74.95	29.19	32.26	32.42

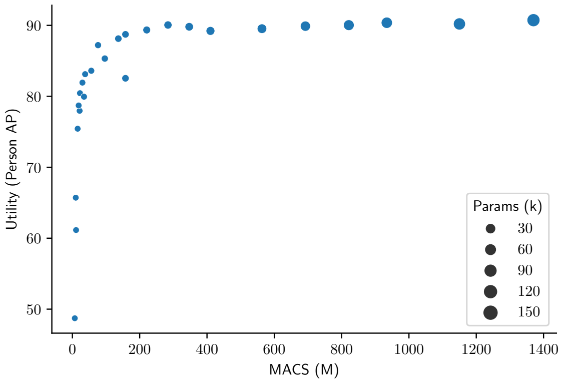

Fig. 5 Influence of model complexity (MACS and parameters) on utility. Smaller models lead to more loss in utility.

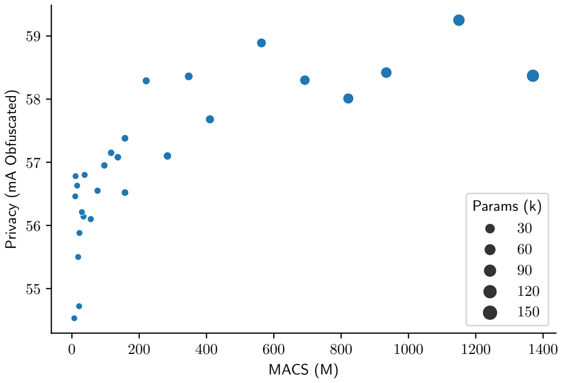

Fig. 6 Influence of model complexity (MACS and parameters) on privacy (the accuracy of a PAR model trained on obfuscated data). Smaller models lead to worse PAR and, thus, better privacy protection.

Consequently, using model complexity to tune the tradeoff may not be ideal.

5 Conclusion and future work

This paper presented a novel framework for privacy-preserving visual analysis in an edge-cloud setup, with pedestrian detection as the example task. The proposed method uses adversarial training to achieve obfuscation of frames while retaining utility without modifying the utility network or requiring labelled sensitive data. Our technique results in near-random PAR performance while maintaining a pedestrian detection accuracy of 90%, outperforming classic and deep learning based obfuscation techniques. The obfuscator is smaller than object detection models and can protect across multiple cameras, making it suited for edge deployment. However, our approach requires white-box access to a trained object detection model and its loss function, limiting its applicability in certain situations. In addition, the obfuscator is reliant on using the object detection model it was trained with.

Future work should focus on developing privacy metrics that do not require ground truth labels and investigating methods to make our technique more object detection model independent. We should also explore the compatibility of our approach in trusted execution environments to ensure its effective deployment on edge devices. Finally, it would prove valuable to research this technique in the setting of smart cars involving non-stationary cameras to understand its performance in more challenging situations.

Declarations

Funding

Sander De Coninck receives funding from the Special Research Fund of Ghent University under grant no. BOF22/DOC/093. This research received funding from the Flemish Government under the "Onderzoeksprogramma Artificiele Intelligentie (AI) Vlaanderen" programme.

Data availability and access

The Avenue dataset is available at http://www.cse.cuhk.edu.hk/leojia/projects/detectabnormal/dataset.html. The ShanghaiTech Campus dataset is available at https://svip-lab.github.io/dataset/campus_dataset.html. The WILDTRACK dataset is available at https://www.epfl.ch/labs/cvlab/data/data-wildtrack/.

Ethical and informed consent for data used

All data used came from public datasets. No additional personal data was collected.

Competing interests

The authors have no competing interests to declare that are relevant to the content of this article.

Author contributions

Sander De Coninck: Conceptualization, Methodology, Investigation, Software, Writing – Original Draft. **Wei-Cheng Wang**: Conceptualization, Writing – Review & Editing. **Sam Leroux**: Conceptualization, Writing – Review & Editing. **Pieter Simoens**: Conceptualization, Writing – Review & Editing, Supervision

References

[1] What's Wrong With Public Video Surveillance? American Civil Liberties Union (2002)

[2] Armstrong, G., Norris, C.: The Maximum Surveillance Society: The Rise of CCTV. Routledge, ??? (2020)

[3] Feldstein, S.: The Global Expansion of AI Surveillance vol. 17. Carnegie Endowment for International Peace Washington, DC, ??? (2019)

[4] Lee, T., Lin, Z., Pushp, S., Li, C., Liu, Y., Lee, Y., Xu, F., Xu, C., Zhang, L., Song, J.: Occlumency: Privacy-preserving remote deep-learning inference using sgx. In: The 25th Annual International Conference on Mobile Computing and Networking, pp. 1–17 (2019)

[5] Yuan, D., Zhu, X., Mao, Y., Zheng, B., Wu, T.: Privacy-preserving pedestrian detection for smart city with edge computing. In: 2019 11th International Conference on Wireless Communications and Signal Processing (WCSP), pp. 1–6 (2019). IEEE

[6] Bentafat, E., Rathore, M.M., Bakiras, S.: Towards real-time privacy-preserving video surveillance. Computer Communications **180**, 97–108 (2021)

[7] Hukkelås, H., Lindseth, F.: Deepprivacy2: Towards realistic full-body anonymization. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 1329–1338 (2023)

[8] Kunchala, A., Bouroche, M., Schoen-Phelan, B.: Towards a framework for privacy-preserving pedestrian analysis. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 4370–4380 (2023)

[9] Huang, C., Kairouz, P., Chen, X., Sankar, L., Rajagopal, R.: Context-aware generative adversarial privacy. Entropy **19**(12), 656 (2017)

[10] Wang, X., Zheng, S., Yang, R., Zheng, A., Chen, Z., Tang, J., Luo, B.: Pedestrian attribute recognition: A survey. Pattern Recognition **121**, 108220 (2022)

[11] Dwork, C.: Differential privacy: A survey of results. In: International Conference on Theory and Applications of Models of Computation, pp. 1–19 (2008). Springer

[12] Doan, T.V.T., Messai, M.-L., Gavin, G., Darmont, J.: A survey on implementations of homomorphic encryption schemes. The Journal of Supercomputing, 1–42 (2023)

[13] Knott, B., Venkataraman, S., Hannun, A., Sengupta, S., Ibrahim, M., Maaten, L.: Crypten: Secure multi-party computation meets machine learning. Advances in Neural Information Processing Systems **34**, 4961–4973 (2021)

[14] Mireshghallah, F., Taram, M., Vepakomma, P., Singh, A., Raskar, R., Esmaeilzadeh, H.: Privacy in deep learning: A survey. arXiv preprint arXiv:2004.12254 (2020)

[15] Taghavi, S., Shi, W.: Edgemask: An edge-based privacy preserving service for video data sharing. In: 2020 IEEE/ACM Symposium on Edge Computing (SEC), pp. 382–387 (2020). IEEE

[16] Himmi, S., Ilter, O., Pailleau, F., Siegwart, R., Bescos, B., Cadena, C.: Don't share my face: Privacy preserving inpainting for visual localization. In: 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 12506–12511 (2022). IEEE

[17] Hukkelås, H., Mester, R., Lindseth, F.: Deepprivacy: A generative adversarial network for face anonymization. In: International Symposium on Visual Computing, pp. 565–578 (2019). Springer

[18] McPherson, R., Shokri, R., Shmatikov, V.: Defeating image obfuscation with deep learning. arXiv preprint arXiv:1609.00408 (2016)

[19] Wang, W.-C., De Coninck, S., Leroux, S., Simoens, P.: An opt-in framework for privacy protection in audio-based applications. IEEE Pervasive Computing **21**(4), 17–24 (2022)

[20] Leroux, S., Verbelen, T., Simoens, P., Dhoedt, B.: Privacy aware offloading of deep neural networks. In: ICML2018, Privacy in Machine Learning and Artificial Intelligence Workshop, pp. 1–3 (2018)

[21] Chan, A.B., Liang, Z.-S.J., Vasconcelos, N.: Privacy preserving crowd monitoring: Counting people without people models or tracking. In: 2008 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–7 (2008). IEEE

[22] Kieu, M., Bagdanov, A.D., Bertini, M., Del Bimbo, A.: Domain adaptation for privacy-preserving pedestrian detection in thermal imagery. In: Image Analysis and Processing–ICIAP 2019: 20th International Conference, Trento, Italy, September 9–13, 2019, Proceedings, Part II 20, pp. 203–213 (2019). Springer

[23] Yang, H., Zhou, Q., Ni, J., Li, H., Shen, X.: Accurate image-based pedestrian detection with privacy preservation. IEEE Transactions on Vehicular Technology **69**(12), 14494–14509 (2020)

[24] Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing **13**(4), 600–612 (2004)

[25] Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 586–595 (2018)

[26] Ding, K., Ma, K., Wang, S., Simoncelli, E.P.: Comparison of full-reference image quality models for optimization of image processing systems. International Journal of Computer Vision **129**, 1258–1281 (2021)

[27] Sheikh, H.R., Bovik, A.C.: Image information and visual quality. IEEE Transactions on image processing **15**(2), 430–444 (2006)

[28] Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017)

[29] Cartucho, J., Ventura, R., Veloso, M.: Robust object recognition through symbiotic deep learning in mobile robots. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2336–2341 (2018)

[30] Lu, C., Shi, J., Jia, J.: Abnormal event detection at 150 fps in matlab. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2720–2727 (2013)

[31] Liu, W., Luo, W., Lian, D., Gao, S.: Future frame prediction for anomaly detection–a new baseline. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6536–6545 (2018)

[32] Chavdarova, T., Baqué, P., Bouquet, S., Maksai, A., Jose, C., Bagautdinov, T., Lettry, L., Fua, P., Van Gool, L., Fleuret, F.: Wildtrack: A multi-camera hd dataset for dense unscripted pedestrian detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5030–5039 (2018)

[33] Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems **28** (2015)

[34] Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)

[35] Jia, J., Huang, H., Chen, X., Huang, K.: Rethinking of pedestrian attribute recognition: A reliable evaluation under zero-shot pedestrian identity setting. arXiv preprint arXiv:2107.03576 (2021)

[36] Cheng, X., Jia, M., Wang, Q., Zhang, J.: A simple visual-textual baseline for pedestrian attribute recognition. IEEE Transactions on Circuits and Systems for Video Technology **32**(10), 6994–7004 (2022)

[37] Specker, A., Cormier, M., Beyerer, J.: Upar: Unified pedestrian attribute recognition and person retrieval. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 981–990 (2023)

[38] He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2961–2969 (2017)

[39] Sovrasov, V.: Ptflops: a Flops Counting Tool for Neural Networks in Pytorch Framework. https://github.com/sovrasov/flops-counter.pytorch