INSPIRE: Instance-level Privacy-preserving Transformation for Vehicular Camera Videos

Zhouyu Li, Ruozhou Yu, Anupam Das, Shaohu Zhang, Huayue Gu,
Xiaojian Wang, Fangtong Zhou, Aafaq Sabir, Dilawer Ahmed, Ahsan Zafar
North Carolina State University, Raleigh, NC, USA

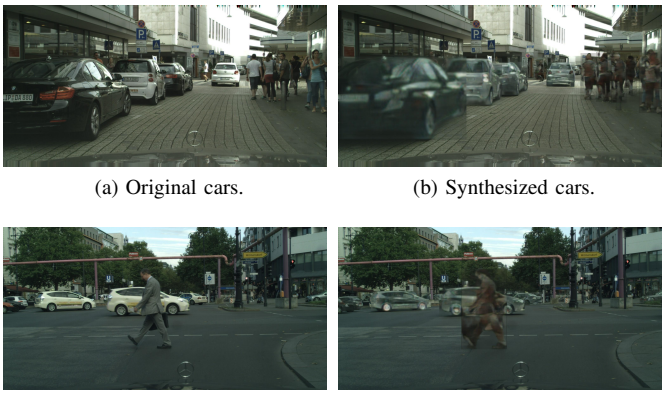
***Abstract*—The wide spread of vehicular cameras has raised broad privacy concerns. Ubiquitous vehicular cameras capture bystanders like people or cars nearby without their awareness. To address privacy concerns, most existing works either blur out direct identifiers such as vehicle license plates and human faces, or obfuscate whole video frames. However, the former solution is vulnerable to re-identification attacks based on general features, and the latter severely impacts utility of the transformed videos. In this paper, we propose an INStance-level PrIvacy-pREserving (INSPIRE) video transformation framework for vehicular camera videos. INSPIRE leverages deep neural network models to detect and replace sensitive object instances in vehicular videos with their non-existent counterparts. We design INSPIRE as a modular framework to enable flexible customization of protected instance categories and their protection modules. An implementation of INSPIRE focused on protecting people and cars is described, which we tested on six re-identification datasets and three real-world vehicular video datasets to evaluate its privacy protection and utility preservation capability. Results show that INSPIRE can thwart 97% of re-identification attacks for people and cars while maintaining a 0.75 object detection mean average precision on transformed instances. We also demonstrate experimentally that INSPIRE is robust against model inversion attacks. Compared to solutions that provide comparable privacy protection, INSPIRE achieves relatively 1.76 times higher counting accuracy and 31.61% higher object detection mean average precision.**

***Keywords*—Privacy, vehicular cameras, video processing, vehicular systems**

I. Introduction

Many modern vehicles are equipped with vehicular cameras such as dash cameras and assisted driving cameras. Recent surveys show that the global dashboard camera market is expected to grow at a compound year rate of around 9.5% from 2022 to 2030 [1]. More than 70% of vehicle buyers listed an integrated dash camera as a desired feature [2].

Vehicular cameras constantly record the vehicle's surroundings. Recorded videos can be used for different purposes, such as providing evidence in accident investigation, contributing to online street views, or building autonomous driving datasets [3], [4]. However, these videos, while containing useful information such as accidents and road hazards, may also contain information about bystanders, such as surrounding vehicles and pedestrians. This usually happens without the awareness of the bystanders. Moreover, the lack of communication channels makes it hard for bystanders to "opt out" from the video collection process [5], leading to their privacy concerns [6]. Sharing such vehicular videos can also expose the video owner to legal risks [7]. Unlike stationary surveillance cameras, ubiquitous and highly mobile vehicular cameras cause more discomfort for bystanders, and open the door for large-scale attacks [5], [6]. For instance, an attacker can launch mobile crowd-sensing campaigns [8] to collect videos on a city scale for surveillance and violate individual privacy rights.



(a) Original cars. (b) Synthesized cars.

(c) Original person. (d) Synthesized person.

Fig. 1: Examples of INSPIRE's transformation. (a) and (c) are original video frames. (b) and (d) are video frames transformed by an INSPIRE-based system, which aims to replace every person and car with a non-existent one.

To address the above concerns, a widely used approach is to detect and blur sensitive attributes of video-captured object instances such as human faces or vehicle license plates [3]. Unfortunately, this approach cannot prevent privacy leakage from exposed quasi-identifiers such as human clothes or vehicle stickers, which are usually enough for an informed attacker to identify sensitive object instances [9]–[12]. Moreover, state-of-the-art re-identification (Re-ID) methods [13]–[19] can identify obfuscated instances across frames and cameras by general features extracted with a deep neural network (DNN)-based model, which further weakens attribute-level privacy protection methods. Existing work has also chosen to blur entire video frames to hide sensitive details [20]. However, this frame-level video transformation can significantly reduce the utility of the videos in analytical tasks such as statistical counting or object detection. To improve privacy protection while still enabling video analytics, the protection scope has to be chosen carefully.

In this work, we design an INStance-level PrIvacy-pREserving (INSPIRE) video transformation framework to provide a modular, scalable solution for the privacy protection of vehicular camera videos. As Figure 1 shows, INSPIRE aims to achieve instance-level privacy: instead of obfuscating predefined sensitive attributes of each instance, it aims to fully replace the instance with a non-existent counterpart. An attacker,

Coauthors Li, Yu, Das, Zhang, Wang, Gu, Zhou, Sabir, Ahmed, Zafar ({zli85, ryu5, anupam.das, szhang42, hgu5, xwang244, fzhou, asabir2, dahmed2, azafar2}@ncsu.edu) are all with NC State University, Raleigh, NC 27606, USA. This work was supported in part by NSF grants 2045539 and 2007391. Information reported here does not reflect the position or the policy of NSF.

even with some prior knowledge and/or state-of-the-art Re-ID models, cannot unveil the identity of the replaced instance because all its identifiable attributes are hidden by replacement. Meanwhile, INSPIRE also aims to achieve a high utility of the transformed video, with minimal degradation of performance when the video is used for common analytical tasks such as statistical counting or object detection. To achieve these goals, we propose a deep learning-based pipeline, where DNN models are used to detect and segment sensitive instances, and *generative adversarial network (GAN)* models are used to synthesize non-existent instances for replacement.

Following the pipeline, we implemented an INSPIRE system to replace every person or car [1] in videos with a non-existent counterpart at the same position with the same contour. Using six Re-ID datasets and three real-world vehicular camera datasets, we comprehensively evaluated our system's utility-preserving and privacy-protection performance. Results show that a well-configured INSPIRE system can thwart over 97% of Re-ID attacks on its transformed instances while maintaining 0.75 mean average precision (mAP) on object detection tasks across different datasets. Attempts of model inversion attacks are also thwarted by the design of the INSPIRE framework. Compared with other systems that thwart around 90% Re-ID attacks on transformed instances, INSPIRE improves the statistical counting accuracy by 1.76 times and maintains 31.61% higher mAP for object detection.

We summarize this paper's contributions as follows:

- We propose an instance-level privacy-preserving video transformation framework called INSPIRE that achieves strong privacy protection and high utility preservation for vehicular camera videos.
- We build an INSPIRE system with car and person as protected categories, which adopts advanced DNNs and enables scalable implementation on commodity hardware.
- We evaluate the implemented system on multiple datasets and show its superior privacy-utility trade-off compared to the state-of-the-art video privacy protection mechanisms.

The rest of the paper is organized as follows: In Section II, we introduce related works. In Section III, we outline relevant techniques used to construct and implement our framework. In Section IV, we present our threat model. In Section V, we introduce the design of the INSPIRE framework and an implementation for person and car protection. In Section VI, we evaluate the privacy protection and utility preservation performance of our implemented INSPIRE system on different datasets. In Section VII, we conclude this paper.

II. Related Works

Various methods have been proposed to protect the privacy of vehicular camera videos from different scopes, including attribute-level, instance-level, and frame-level protection.

Attribute-level protection. One widely used technique is to blur human faces and vehicle license plates [3]. To make a balance between privacy and utility, Yu *et al.* [22] proposed a GAN-based method to replace human faces and vehicle license plates with machine-synthesized ones and introduced differential privacy to the synthesis process. Fan *et al.* [23] searched for the optimal blurring level by formulating an optimization problem. However, given some prior knowledge, an adversary can still identity individuals with their exposed attributes [9], [10]. State-of-the-art Re-ID models [13]–[19] can be used to either identify the same object across multiple videos or compare blurred objects with ground truth for identity inference, both violating the privacy of identified instances.

Instance-level protection. Since privacy protection on empirically defined sensitive attributes is not secure, researchers have made attempts on instance-level privacy protection. Uittenbogaard et al. [24] designed a Multi-view Inpainting Network to remove an entire protected instance by combining images taken from different perspectives. However, this approach requires users to have images from different perspectives, and the sanitized image can no longer be used for tasks like statistic counting. Nodari et al. [25] proposed to replace each pedestrian in street view images with another counterpart from an authorized dataset. But a small authorized dataset makes it hard to find a suitable counterpart for every pedestrian, while expanding the authorized dataset can be expensive and may have other privacy and legal concerns.

Frame-level protection. Recent efforts have tried to preserve privacy and utility by reversibly transforming the whole video frame into a vague style [26]. However, the reversibility makes the system vulnerable to model inversion attacks, and blurring the whole frame impacts nonsensitive parts of the video frames, which may contain useful information.

Advantages of INSPIRE. Compared to attribute-level protections [3], [22], INSPIRE focuses on efficiently hiding all the attributes of protected instances. Compared to works obfuscating whole video frames [20], INSPIRE only influences protected instance areas to maintain video utility for analytical tasks. Compared to other instance-level privacy-protection works [24], [25], replacing with synthesized data allows INSPIRE to have an unlimited replacement data source and have no real data exposed.

III. Background

In this section, we discuss preliminary tools used for either privacy attack or defense in this paper.

Object detection. Object detection is a computer vision task that finds specific kinds of instances in digital images [27]–[29]. An object detection system takes an image as the input and outputs a bounding box (object location), a class index (object category), and, optionally, a confidence value for each detected object. YOLO (You Only Look Once) [29] is a set of real-time object detection models that only passes input data through its network once. The recent YOLO models (e.g., YOLOv5 [30]) can achieve both high accuracy and high efficiency in most object detection scenarios, making it one of the most prevalently used models in real-world applications.

Semantic segmentation. Image semantic segmentation is a pixel-level classification task on a digital image. A segmentation model outputs a segmentation mask of the same size as the original image, with the value of each pixel representing the classification result. Commonly used semantic segmentation models include UNet [31] and DeepLabV3 [32]. Short paths between symmetric layers of UNet [31] allow the decoder to access condensed and raw image features for fine-grained segmentation, and we apply UNet in our system implementation.

[1] The category taxonomy is according to the Coco dataset [21].

Generative adversarial network (GAN). GAN [33] is an unsupervised generative machine learning framework. A GAN consists of two deep learning models: a generator and a discriminator. The generator takes random inputs and synthesizes images that are indistinguishable from the real ones. On the other hand, the discriminator tries to distinguish the images synthesized by the generator from real-world images. The two models are updated alternatively in the training phase. After training, the generator is expected to generate images indistinguishable from real-world images. GAN has achieved superior performance in many image synthesis tasks [33]–[35]. Once trained, the basic GAN models do not allow further customization of the synthesis process without retraining. Conditional GAN (CGAN) [36] was proposed to give users more control in the synthesis process. In CGAN, the generator model synthesizes non-existent images according to condition labels. Pix2pix [36] is a variant of CGAN where the discriminator examines the input image pixel by pixel. Based on Pix2pix, Pix2pixHD [37] was proposed to improve the quality of synthesized images. We leverage the Pix2pixHD [37] model to implement the image generators in our system.

Re-identification (Re-ID). Re-ID systems find images containing the same object as the object in a given query image. A Re-ID system requires the user to have a gallery dataset that contains images of different instances taken from different angles or using different cameras. Given the query image of a specific object, the Re-ID system aims to retrieve images of the same object from the gallery dataset. To do this, a Re-ID system first extracts features of all the gallery images with a DNN model. For each query image, the system extracts its feature and computes the distances between the query feature and the features of gallery images. A smaller feature distance indicates a higher probability that the two images contain the same object. With the help of the Re-ID model, an attacker with a large-scale dataset can launch Re-ID attacks to uncover the identities of their instances of interest. This paper considers Re-ID as an attack in our threat model, and in Section VI, we show that existing methods are vulnerable to this Re-ID attack through experiments. We further validate that our proposed framework can effectively thwart this kind of attack.

Model inversion attack. Given some inputs, a deep learning model usually outputs the user-desired results via linear and non-linear data transformations. If an adversary can obtain a large number of input-output pairs of a given deep learning model, the adversary can train another deep learning model to inverse the transformation of the given model. We also consider model inversion attack in our threat model and experimentally show that INSPIRE is secure against this attack.

IV. Threat Model

Trusted and untrusted environments. We assume a system built with INSPIRE is installed in a vehicle's onboard unit (OBU) as a software plugin. The system processes the video before it is transmitted, or shared beyond the in-vehicle storage. Alternatively, it could be implemented on a user's mobile device or desktop with enough computational resources. In either case, *we assume the video content remains private before INSPIRE processes it*. Attacks that happen before and during INSPIRE processing, such as an attacker compromising the vehicle's OBU or raw video transmission to a user's computing device, are assumed to be defended by security measures orthogonal to INSPIRE. After INSPIRE processing, the transformed video is shared with external parties for display or video-based analytics. *Any party with access to the video content after transformation is assumed not to be trusted.*

Adversary's goal and capability. An adversary tries to *reveal the identities of instances in the captured and transformed video*. We assume *the adversary has full knowledge of our framework, except for the random number generator (RNG) used to provide randomness for synthesized images*. Specifically, the training data for each building block can be publicly available and accessible to the adversary. Following the framework, the adversary with enough computational power can build the same system as the user's. We assume the adversary may also have prior knowledge of some replaced instances. The knowledge is in the format of a large-scale dataset with images of instances of interest to the adversary. However, the adversary does not have enough resources to manually check the dataset and compare each image to every transformed video frame. Moreover, the adversary can query the user's INSPIRE system with his or her own videos as many times as needed for an attack. The only confidential data to the adversary is the original video and the RNG for image synthesis.

Attacks. With the above capabilities, the adversary can launch two kinds of attacks: the *Re-ID attack* and the *model inversion attack*. By launching the Re-ID attack, an adversary with state-of-the-art Re-ID models and the large-scale auxiliary dataset will try to identify as many instances of interest as possible in the transformed video. As for the model inversion attack, an adversary will attempt to train another model to reconstruct the original video frames from the transformed one. Considering these attacks, we design our framework in the next section.

V. Framework Design and Implementation

In this section, we first walk through our framework and then introduce details and implementation of each building block, with *person* and *car* as protected categories.

A. Overview

Our INSPIRE framework aims to achieve instance-level replacement of objects in a user-defined protected category list with synthesized counterparts. Moreover, we hope systems built with our framework to have easy-to-customize protected categories and their protection modules. Toward this end, we design a modular privacy-preserving video transformation framework named INSPIRE as shown in Figure 2, which consists of three major building blocks: a global *instance curator*, per-category *feature extractors*, and per-category *generators*.

Before building a system following INSPIRE framework, a list of protected categories should be defined to specify instance categories the system aims to protect. For each system, there exists one global instance curator to detect and replace instances of all protected categories, while every category possesses a pair of feature extractor and generator for segmentation, auxiliary feature extraction, and image synthesis. Prior to deploying the system, users can customize and train object detection model for protected categories. Suitable DNN models should also be trained for every feature extractor and generator to perform semantic segmentation and image synthesis tasks.

In the application phase, to thwart Re-ID attacks across video frames, INSPIRE processes the video as a frame queue.

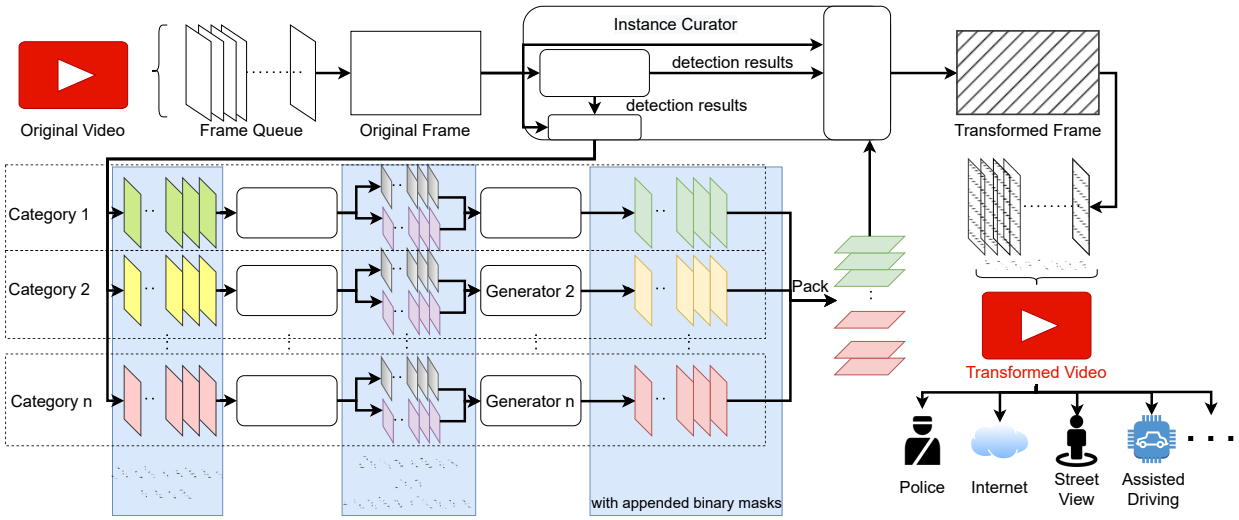


Fig. 2: INSPIRE framework: Instance curator detects, crops, and divides instances on the video frame according to their categories. Each category has a pair of feature extractor and generator to synthesize non-existent instances, which are then patched back to the video frame by the instance curator.

(a) Cropped (b) Single (c) Binary (d) Auxiliary (e) Synth.

Fig. 3: Processing outcomes of INSPIRE modules for an instance. (a) Cropped: cropped image in the instance curator's crop method before padding and scaling. (b) Single: single-instance image in the instance curator's crop method after padding and scaling. (c) Binary: binary mask given by the feature extractor. (d) Auxiliary: auxiliary (edge) feature given by the feature extractor. (e) Synth.: synthesized-instance image given by the generator.

Distinct non-existent counterparts will replace the same instance on consecutive video frames to disentangle the cross-frame information. For each frame, the instance curator's *detect* method will first be called to detect instances of protected categories. Next, the instance curator's *crop* method will crop out each protected instance according to its bounding box (Figure 3(a)), and resize and pad it as a *single-instance image* (Figure 3(b)). *Single-instance images* of different categories will be divided, and the ones of the same category will be packed into a batch and fed into its feature extractor to get a batch of *binary masks* (Figure 3(c)), isolating the areas of instances from backgrounds, and their *auxiliary features* (Figure 3(d)), providing spatial information to facilitate image synthesis. Then the category's generator will take *binary masks* and *auxiliary features* as conditions to generate non-existent *synthesized-instance images* (Figure 3(e)) of the same category that precisely fill *binary masks*' instance areas and follows *auxiliary features*'s spatial outline. Different categories' feature extraction and generation processes can run in parallel to accelerate the transformation. Finally, the instance curator will collect *synthesized-instance images* of different categories and patch them back to the frame with the *patch* method to finish the transformation. The transformed frame will be appended to the tail of the transformed video queue and the system will intake the next frame until the end of the video.

Customizibility. A manufacturer can define a protected category list with some commonly used categories and prepare the initialization process for users. Only one model is needed to be trained for each building block, and well-trained models can be copied and deployed in each ex-factory system. As every building block can be trained independently and plugged into the framework without affecting other building blocks, this modular design eases the customization of protected categories and supports upgrades where newer and faster DNN models can be plugged-and-played. To facilitate this customization, we defined application programming interfaces (APIs) for each building block in Table. I, specifying their input and output. Manufacturers or third-party developers can develop customized building blocks following the APIs for additional categories, and use the latest DNN models to improve existing building blocks' performances. On the other hand, users can download and install the released building blocks according to their needs to customize or upgrade their systems.

TABLE I: Inputs and outputs of INSPIRE's three building blocks.

Building blocks		Inputs	Outputs
Instance curator	detect	original video frames	object detection results
	crop	original video frames, object detection results	single-instance images
	patch	binary masks, synthesized-instance images, object detection results	transformed video frames
Feature extractor		single-instance images	binary masks, auxiliary features
Generator		binary masks, auxiliary features	synthesized-instance images

(a) Original (b) Binary mask (c) Instance mask (d) Edge mask

Fig. 4: Overlapped instances synthesized with different auxiliary features.

Why do we need auxiliary features? Though each *single-instance image* is supposed to contain only one instance, overlapping will lead to multiple instances in one *single-instance image*, as shown in Figure 4(a). In this case, *binary mask* alone will let the generator regard the overlapped instances as an instance with abnormal contour and synthesize a single

instance with an irregular shape, as shown in Figure 4(b), which harms the utility of transformed videos. Hence, *auxiliary features* are needed to provide spatial relationships among overlapped instances. In Section V-C, we will discuss more details about *auxiliary features* selection.

B. Instance Curator

An instance curator consists of three main methods: *detect*, *crop*, and *patch*. On each input frame, the *detect* method leverages an object detection model to detect instances of protected categories. Every entry in the detection result has three elements: a bounding box denoting the rectangle area that contains the instance, a class index indicating the instance's category, and a confidence level representing the probability that this entry is correct. The *detect* method then passes the detection result to the *crop* and *patch* method. The *crop* method crops the detected instances out of the video frame according to their bounding boxes and transforms them into *single-instance images* of the same size. Finally, the *crop* method passes every batch of *single-instance images* to its feature extractor-generator pair to synthesize their same-shape non-existent counterparts for replacement. After all the generators complete synthesizing non-existent instances, the instance curator collects *binary masks* and *synthesized-instance images* from generators and passes them to its *patch* method. Given a *binary mask* M, a *synthesized-instance image* F, and the original *single-instance image* I, the *patch* method fits the synthesized instance into the original background by applying $F \leftarrow F \cdot M + I \cdot (1 - M)$. In this way, the original instance is entirely replaced by the synthesized one due to the first term $F \cdot M$, while the original background is preserved by the second term $I \cdot (1-M)$. Then each *synthesized-instance image* is scaled to its original size according to the detection result and patched back to the region indicated by its bounding box.

Implementation. We used a pre-trained YOLOv5s [30] object detection model for implementing the *detect* method. We rounded down all the decimal coordinates for bounding boxes of detected objects to avoid overflowing the image. In the *crop* method, we cropped each detected instance according to its bounding box and symmetrically zero-padded it into a square image. Instances of the same category were scaled to the same size and stacked into a batch to facilitate parallel processing. The *patch method* fit synthesized instances into the video frame to finish the replacement. Specifically, it patched instances back to the original video frame sequentially according to the ascending order of their bounding box sizes to keep the distance information among different instances.

C. Feature Extractor

Given a batch of *single-instance images*, a feature extractor separates the instances from backgrounds and extracts images' *auxiliary features*. A DNN model achieves instance-background separation by performing semantic segmentations. The outputs are single-channel images called *confidence maps*. The value of each pixel on a *confidence map* denotes the probability that the pixel is part of an instance. After acquiring the *confidence map*, the feature extractor gets the instance's *binary mask* by rounding each value to 0 (denoting a background pixel) or 1 (denoting an instance pixel). Meanwhile, DNN-based or traditional methods are adopted to extract auxiliary features. In practice, a feature extractor takes a batch of *single-instance images* and processes them simultaneously to accelerate this process. In the following paragraph, we first select our *auxiliary feature* and then introduce the implementation of our INSPIRE system's feature extractor.

Auxiliary feature selection. To provide the spatial relationship of overlapped instances, we considered two kinds of *auxiliary features*: *instance-level segmentation masks*, where different instances have different pixel values, and binary *edge masks* with 1 for object edges and 0 for other areas. Compared with *synthesized-instance images*, which slightly alleviate the problem as shown in Figure 4(c), the image synthesized with *edge mask* as the *auxiliary feature*, shown in Figure 4(d), successfully synthesized overlapped instances. So we selected *edge mask* as our system's *auxiliary feature*.

Implementation. For semantic segmentation, we trained simplified UNet [31] models, whose first layer dimension was reduced from 64 to 8, and the reduction proportionally propagated to the following layers. The simplified model was 63 times smaller than the standard one while still producing satisfiable *binary masks*. To train the models, we built a single-instance binary semantic segmentation dataset based on the Cityscapes dataset [38] by applying the instance curator's *crop* method on original and segmentation images according to the ground-truth bounding boxes. The dataset contained 3558 people and 9948 cars. Each model was trained for 200 epochs on the dataset with the Adam optimizer whose learning rate lr and parameters β_1, β_2 were set as $lr = 0.0002$, $\beta_1 = 0.9$, $\beta_2 = 0.999$. We used L1 distance as the loss function, and the batch size was set as 4. For edge feature extraction, we leveraged the *Canny* edge detection algorithm [39]. As *edge mask* might contain detailed information of interest to the adversary, we applied a Gaussian filter to blur the *single-instance images* before edge detection. A Gaussian filter has two parameters: kernel size and standard deviation (SD), where a higher kernel size and a higher SD cause heavier blurring effects on applied images. Users can tweak these two parameters to adjust the amount of edge information exposed through *edge masks* to make a privacy-utility trade-off.

D. Generator

For each protected category, a generator is used to synthesize same-category non-existent instances. A conditional GAN (CGAN) model is adopted for this conditional image synthesis task. For every instance, the model will synthesize a *synthesized-instance image* conditioned on its *binary mask* and *auxiliary feature*. In practice, similar to the feature extractor, the generator also parallelly processes a batch of *binary masks* and *auxiliary features* from the same category. To provide randomness, *random latents* are concatenated with *binary masks* and *auxiliary features* as the input of each generation model to generate *synthesized-instance images*.

Implementation. We trained Pix2pixHD [40] models to synthesize non-existent instances conditioned with *binary masks* and *edge masks*. In the training dataset, *binary masks* and *single-instance images* were from the binary semantic segmentation dataset built in Section V-C, and *edge masks* were acquired by applying the Canny algorithm on *single-instance images*. Each Pix2pixHD model was trained for 200 epochs with default hyperparameters [40].

(a) Original (b) INSPIRE (c) Dashcam Cleaner (d) BBox Blur (e) SecGAN

Fig. 5: Transformed frames from compared systems

VI. Evaluation

A. Re-ID Attack

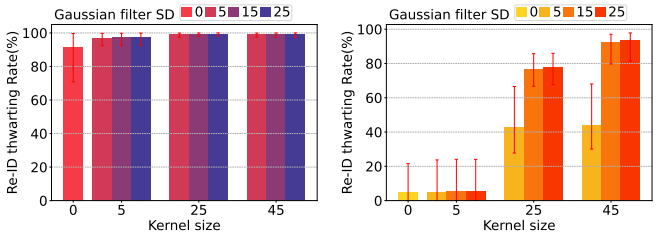
TABLE II: Details about Re-ID datasets

Name	Query images	Gallery images	Gallery instances	Category	Real world
Cityscapes (person)	4924	4924	267	person	✓
Duck MTMC	2228	17661	1110	person	✓
Market-1501	3368	19732	752	person	✓
Cityscapes (car)	10450	10450	147	car	✓
VeRi	1678	11579	200	car	✓
VeRi-CARLA	424	3823	50	car	✗

Evaluation data. We evaluate our system on three *person* Re-ID datasets and three *car* Re-ID datasets, as shown in Table II. For each category, we have two widely used datasets and one self-made dataset built from the Cityscapes demo videos[2]. Specifically, we use the Market-1501 [41] and Duke MTMC [42] datasets for *person* Re-ID evaluation, and the VeRi [14], [19] and VeRi-CARLA [43] datasets for *car* Re-ID evaluation. To get gallery images of Cityscapes Re-ID datasets, people and cars in each video frame were cropped according to their bounding boxes. Images for the same instance were linked with an object tracking model [44]. Query images were directly copied from the gallery images. This self-made dataset simulates the strongest Re-ID attacks where the adversary has the exact original images for transformed instances and would like to re-identify each transformed instance with respect to the original one. In practice, an adversary would generally not have the exact original images for the attack, thus leading to weaker attacks than what has been evaluated with this dataset.

Compared systems. We compare our INSPIRE system with three privacy-preserving video transformation systems: *Dashcam Cleaner*, *BBox Blur*, and *SecGAN*. For every compared system, we present the transformed video frame in Figure 5. ***INSPIRE*** (Figure 5(b)) conducts inplace replacement for every person and car in the video frame with a non-existent counterpart. We evaluate INSPIRE with and without Gaussian filters applied before getting the *edge masks*. ***Dashcam Cleaner*** [3], [45] (Figure 5(c)) blurs detected faces and license plates in each video frame. ***BBox Blur*** [46] (Figure 5(d)) directly applies a Gaussian filter on the whole protected instances according to their detection bounding box. To accommodate images smaller than the filter's kernel, each image is resized to 256×256 before applying the filter and resized back after being blurred. We also evaluate a Non-blurring BBox Blur system, which only conducts detection and scaling operations. ***SecGAN*** [20], [47] (Figure 5(e)) is a recent privacy-preserving traffic video transformation system. It translates every video frame into the cartoon style to hide sensitive attributes such as faces and license plates.

[2]Demo videos are orthogonal to our model training images in Section V.



(a) INSPIRE (b) BBox Blur

Fig. 6: Average Image-wise Re-ID thwarting rates for INSPIRE and BBox Blur with different kernel sizes and SDs of applied Gaussian filters. Error bars are the min-max fluctuation on different datasets.

Adversarial systems. We run the state-of-the-art person Re-ID system [48] and car Re-ID system [49], [50] to evaluate compared systems' performances against Re-ID attacks. We use the pre-trained weight for the person Re-ID system, which has 88.84% accuracy[3] on the Market-1501 [41] dataset. We also use the pre-trained weight for the car Re-ID system, which has 96.7% accuracy on the VeRi [14], [19] dataset. Though Market-1501 and VeRi datasets were used to train our adversary systems, in evaluation, we only use their test sets, which are also orthogonal to their training sets.

Experiments. We transform the query images in every dataset with compared systems. We query each transformed image with the Re-ID model for its specific category and recorded the identity of the top-ranked gallery image.

Metrics. For every compared system, on each dataset, we calculate the image-wise Re-ID thwarting rate, which is the percentage of the unidentified or wrongly-identified query images over the total number of query images.

Influence of the Gaussian filter on privacy. We first inspect the influence of the applied Gaussian filter on INSPIRE and BBox Blur's Re-ID thwarting rates. We compare the image-wise Re-ID thwarting rate with varying Gaussian filter kernel sizes and SDs. Combinations between three kernel sizes $\{5, 25, 45\}$ and three SDs $\{5, 15, 25\}$ are selected. Figure 6 shows the average image-wise Re-ID thwarting rate of the two compared systems over different datasets with different Gaussian filters applied. For each system, the min-max fluctuations on different datasets are presented with error bars on each data bar. The bar with zero kernel size and zero SD (the leftmost one) means no Gaussian filter is applied in the system. We have the following findings from the inspection.

> Applying the Gaussian filter in INSPIRE can improve and stabilize the protection performance against Re-ID attacks.

In both Figure 6(a) and Figure 6(b), compared to data

[3]Here we use the Rank-1 accuracy, measuring the percentage of top-ranked instance identities given by the Re-ID system to match identities of the query images. The benchmark data is based on the original dataset.

bars without a Gaussian filter, the ones with Gaussian filters are higher, and their error bars are also significantly shorter. This implies that applying a Gaussian filter can improve and stabilize the system's performance in thwarting Re-ID attacks.

For INSPIRE and BBox Blur, improving the kernel size and SD of the Gaussian filter enhances the Re-ID thwarting rate.

In Figure 6(a) and Figure 6(b), Re-ID thwarting rates of both INSPIRE and BBox Blur increase as the kernel size and SD of applied Gaussian filter increase. And from Figure 6(b) we find that kernel size and SD mutually upper bound the BBox Blur's Re-ID thwarting rate.

In INSPIRE, applying a Gaussian filter with small kernel size and SD is sufficient to thwart most Re-ID attacks.

Figure 6(a) shows INSPIRE with a minor blurring Gaussian filter (kernel size 5, SD 5) has already raised the worst Re-ID thwarting rate to over 97%. Higher kernel size and SD brings INSPIRE's Re-ID thwarting rate to around 99%. This indicates that applying a Gaussian filter with a small kernel size and SD is enough to thwart almost all the Re-ID attacks.

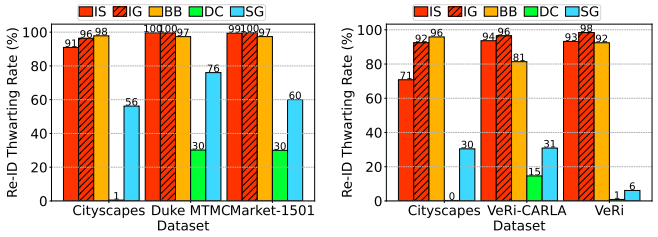
Comparison with different systems. According to the discussion about Gaussian filters, in our rest comparison, for INSPIRE, we consider INSPIRE without a Gaussian filter named **INSPIRE** and INSPIRE with a minor blurring Gaussian filter (kernel size 5, SD 5) named **INSPIRE-Gaussian**. For BBox Blur, we consider the heaviest blurring Gaussian filter (kernel size 45, SD 25) named **BBox Blur**. Figure 7 shows compared systems' image-wise Re-ID thwart rates on different datasets, from which we have the following insights.

In practice, INSPIRE with Gaussian filter can effectively thwart Re-ID attacks for its protected instances.

In the results of both person Re-ID attacks (Figure 7(a)) and car Re-ID attacks (Figure 7(b)), data bars for INSPIRE-Gaussian (red shadow bars) are higher than almost all the compared systems on different datasets, except that on the two Cityscapes datasets it is two to four percent lower than BBox Blur (yellow bars). However, they both achieve over 90% Re-ID thwarting rate. Any system at that thwarting rate has already left the attacker's advantage similar to a random guess. Meanwhile, INSPIRE without a Gaussian filter (red bars) also achieves at least 90% Re-ID thwarting rates on all the datasets, except for the 70.8% thwarting rate for car Re-ID attacks on the Cityscapes dataset. But the two Cityscapes datasets simulate the strongest attacker with the exact original copy of each query image before the transformation, which is unlikely to happen in actual attacks. The above observations imply that INSPIRE can thwart most Re-ID attempts in practice and generalizes well across a wide range of datasets.

Attribute-level and frame-level obfuscation cannot thwart Re-ID attacks with state-of-the-art deep learning models.

The highest person and car Re-ID thwarting rates for Dashcam Cleaner (green bars) are 30.2% and 15%, on the Duke MTMC dataset and VeRi-CARLA dataset, respectively. However, vehicles in the VeRi-CARLA dataset come from the CARLA simulator instead of the real world. On the other two real-world vehicle datasets, Re-ID thwarting rates for the Dashcam Cleaner are almost zero. For SecGAN (blue bars),



(a) Person Re-ID thwarting rates (b) Car Re-ID thwarting rates

Fig. 7: Compared systems' image-wise Re-ID thwarting rates on different datasets. (IS: INSPIRE, IG: INSPIRE-Gaussian, BB: BBox Blur, DC: Dashcam Cleaner, SG: SecGAN)

though the person Re-ID thwarting rates are above 50% on all the datasets, none of them reaches 80%, and its car Re-ID thwarting rates are smaller than 31%. This indicates that blurring pre-defined sensitive attributes like faces and license plates cannot protect instance privacy against Re-ID attacks as other attributes are still exposed and can be captured by Re-ID models, while blurring the whole video frame may obfuscate some details but still exposes the general features of objects.

B. *Model inversion Attack*

Intuition. Another privacy concern regarding INSPIRE is attacks targeting on its DNN models, such as the model inversion attack. Intuitively we deem INSPIRE safe from model inversion attack due to its two-stage design. The feature extractor has removed all the texture information of the original instance. The instance contour and spatial outline, which are from its *binary mask* and *auxiliary features* and not intended to be hidden, are the only information passed to the generator and hence flow to the transformed video. Even having white-box access to the model, the transformed video is free of sensitive information from the removed original instances.

Experiment setup. We design and conduct an experiment to validate our intuition. We assume attackers can query the video transformation system repetitively to get a dataset with enough original and transformed video frames, and use the dataset to train inversion models to reverse the transformation. In the experiment, we launched the model-inversion attack on two DNN-based systems: SecGAN and INSPIRE. No Gaussian filter was applied in INSPIRE. We collected 9948 transformed-original image pairs for each system by querying the system with single-instance car images in the Cityscapes training dataset. Pix2pixHD models with the same setup as INSPIRE generators were used as our inversion models. We trained every inversion model for 200 epochs with transformed images as inputs and original images as labels. The evaluation was conducted on the Cityscapes test dataset. From the evaluation result, we validate that:

Fig. 8: An example of model inversion attack. The license plate number is "CR2:EE:17", which can be recognized in the original (1st) and SecGAN inverse (3rd) images, but not the rest.

TABLE III: Details about utility evaluation datasets.

Dataset Names		Number of videos	Average people per frame	Average cars per frame
Cityscapes		3	5.70	4.68
Accident	Positive	17	2.08	4.45
	Negative	31	2.60	4.82
BDD100K		54	0.95	4.04

INSPIRE can thwart model inversion attacks by design.

Figure 8 shows an example of the evaluation result. We can find that attributes like vehicles' license plates are obfuscated in images transformed by SecGAN and INSPIRE. However, as shown in the third column of Figure 8, the license plate number obfuscated by SecGAN can be reconstructed by its inversion model and become recognizable again. On the contrary, the fifth column of Figure 8 shows that the inversion attempt failed on instances obfuscated by INSPIRE.

C. Video Analytics Utility of Transformed Videos

Evaluation datasets. We evaluate compared systems utility-preserving performance on three vehicular video datasets: Cityscapes demo videos, BDD100K dataset [51], and Dashcam Accident dataset [52]. Evaluation videos comprehensively contain various scenarios, including accidents and regular drives, days and nights, and different weather conditions. Details for evaluation datasets are listed in Table III.

Metrics. We evaluate the transformed video's utility for two video analytic tasks: statistical counting and object detection. For statistical counting, we compute the *counting accuracy* for each video, which is one minus the mean absolute counting errors across all the frames divided by the average number of objects on each frame. For object detection, we compute the mean average precision (mAP) of the detection results on transformed videos. As a widely used evaluation metric for object detection models, mAP jointly reflects detection precision and recall in the range of $[0, 1]$. Detection results on original videos are regarded as the ground truth.

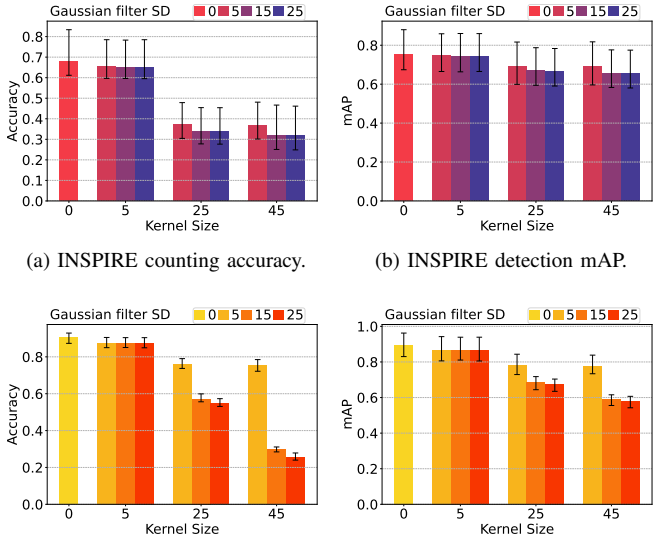
Comparative systems. Utility evaluation uses the same compared systems as the Re-ID attack evaluation in Section VI-A.

Experiment setup. we first collected transformed videos by applying compared systems to original videos. Then we fed the original and transformed videos into a YOLOv5m object detection model and recorded each detected person and car's bounding box, class index, and confidence level. We adopted the more complicated YOLOv5m model instead of the YOLOv5s used in INSPIRE to offer a fair comparison. The confidence threshold of the YOLOv5m model was set as 0.5, and the rest parameters were kept as default.

Influence of Gaussian filter on utility. Figure 9 shows the influence of applied Gaussian filters on the utility of videos transformed by INSPIRE and BBox Blur, from which we have the rule-of-thumb for selecting INSPIRE's Gaussian filter.

A Gaussian filter with a small kernel size and a small SD is recommended to be applied in INSPIRE.

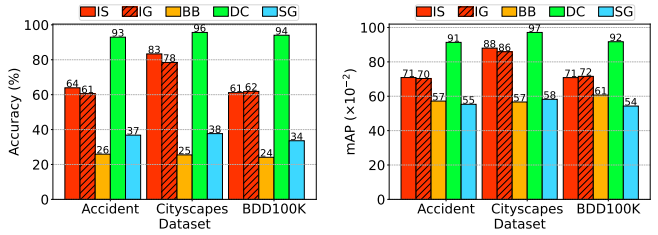
In Figure 9, we find the minor blurring Gaussian filter (kernel size 5, SD 5) does not significantly degrade the transformed video's utility. According to the Re-ID attack evaluation in Section VI-A, such a Gaussian filter can thwart most Re-ID attacks. Hence, applying a Gaussian filter with a small kernel size and a small SD is recommended for INSPIRE.



(a) INSPIRE counting accuracy. (b) INSPIRE detection mAP.

(c) BBox Blur counting accuracy. (d) BBox Blur detection mAP.

Fig. 9: Counting accuracy and detection mAP of INSPIRE and BBox Blur subject to different kernel sizes and SDs of applied Gaussian filters. Error bars are the min-max fluctuation on different datasets.



(a) Counting accuracy (b) Detection mAP

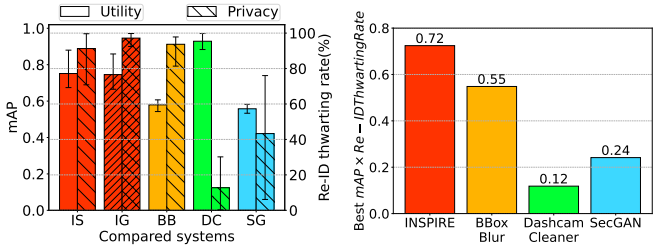
Fig. 10: Counting accuracy and detection mAP of compared systems on different datasets. (IS: INSPIRE, IG: INSPIRE-Gaussian, BB: BBox Blur, DC: Dashcam Cleaner, SG: SecGAN)

The utility of BBox Blur transformed videos decreases as the kernel size and SD of its applied Gaussian filter increase.

In Figure 9(c) and Figure 9(d), counting accuracy and detection mAP of BBox Blur's transformed videos decrease as the kernel size and SD of its applied Gaussian filter increased. Moreover, data bars for BBox Blur without Gaussian filter (left-most bars) still suffer 9.5% accuracy loss and 0.11 mAP loss, though only scaling operation was executed. This implies:

Small perturbations can cause non-negligible impacts on the detection results of the YOLO object detection model.

Comparison with different systems. The counting accuracy and detection mAP of compared systems' transformed videos from different datasets are shown in Figure 10(a) and Figure 10(b), respectively. Since Dashcam Cleaner did not distort or replace each object but hid its sensitive attributes, it has minimal impact on the counting and detection performance and serves as our baseline for the other methods. INSPIRE has higher counting accuracy and detection mAP on the Cityscapes dataset. INSPIRE achieves 83% statistical counting accuracy and 0.88 object detection mAP, while INSPIRE-



(a) Utility-privacy trade-off. (b) Utility-privacy product.

Fig. 11: Utility-privacy trade-off. (a) presents the utility(mAP) and privacy(Re-ID thwarting rate) of compared systems side-by-side. Error bars are the min-max fluctuation on different datasets.(IS: INSPIRE, IG: INSPIRE-Gaussian, BB: BBox Blur, DC: Dashcam Cleaner, SG: SecGAN) (b) shows the utility-privacy product of compared systems. For INSPIRE and BBox Blur, we show the max-achievable product of different applied Gaussian filters.

Gaussian suffers only 5% and 0.02 decrement on counting accuracy and detection mAP, respectively. This is because its image synthesis models were trained on the training set of the Cityscapes dataset, and the tones of synthesized instances are consistent with the background, which causes less perturbation on the original video frame and leads to less influence on the object detection model. The influence can be addressed by fine-grained color normalization on different datasets, and we leave it for future works. Meanwhile, INSPIRE, with or without Gaussian filter, has significantly higher counting accuracy and detection mAP than BBox Blur and SecGAN on all the datasets. Videos transformed by the BBox Blur have the lowest accuracy for statistical counting tasks, and videos transformed by the SecGAN have the lowest mAP for object detection tasks. This is because BBox Blur averaged out the instance with the background, which made object detection harder, and SecGAN severely distorted the whole video frame to make the details unrecognizable. Specifically, comparing INSPIRE and BBox Blur, which both have around 90% Re-ID thwarting rate, INSPIRE achieves at least 44.31% counting accuracy and 0.18 higher detection mAP.

Privacy-utility trade-off. We analyze the privacy-utility trade-off of compared systems from two aspects and conclude that

> INSPIRE achieves the best privacy-utility trade-off among compared systems.

In Figure 11(a), we jointly compare the privacy-utility trade-off for compared systems. We use the average mAP as the utility metric and quantify privacy with the average image-wise Re-ID thwarting rate. Though maintaining the highest utility, videos transformed by Dashcam Cleaner are vulnerable to Re-ID attacks. Compared to BBox Blur and SecGAN, INSPIRE enhances the utility and privacy of transformed videos. Although BBox Blur can improve its utility at the cost of privacy, since its privacy is already lower than INSPIRE, further decreasing the privacy makes it less competitive as a privacy-preserving system.

Figure 11(b) shows the utility-privacy product for compared systems. We define the metric utility-privacy product as the multiplication of a system's mAP and its Re-ID thwarting rate (i.e., Re-ID Thwarting Rate $\times$ mAP). A higher utility-privacy product implies a better privacy-utility trade-off as it requires its two factors to be high at the same time. For INSPIRE and BBox Blur, we present the highest achievable utility-privacy product by applying different Gaussian filters, whose kernel sizes were selected from $\{5, 25, 45\}$ and SDs were selected from $\{5, 15, 25\}$. Figure 11(b) shows INSPIRE has the best utility-privacy trade-off among compared systems. Applying a minor blurring Gaussian filter (kernel size 5, SD 5), INSPIRE achieves the highest utility-privacy product. Specifically, it achieves a 97% Re-ID thwarting rate and a 0.75 detection mAP. The product of 0.724 is 32.11% higher than the second-ranked BBox Blur with heavy blurring Gaussian filter (kernel size 45, SD 15). Though Dashcam Cleaner maintains the most utility, the lack of privacy protection leads to the smallest utility-privacy product 0.12 among compared systems. On the other hand, SecGAN achieves a 0.24 utility-privacy product, which is only one-third of INSPIRE's.

VII. Conclusion

In this paper, we propose an instance-level privacy-preserving video transformation framework called INSPIRE for vehicular camera videos. The framework replaces instances in protected categories with machine-synthesized counterparts in the same shape and at the same location. Identifying information of protected instances would thus be removed with minimal impact on the utility of the transformed video for video analytics tasks such as statistical counting and object detection. Following the framework, we implement a video transformation system to replace people and cars in vehicular camera videos. We trained UNet and Pix2pixHD models as system building blocks for extracting protected instances and synthesizing non-existent instances. We introduce auxiliary features to resolve the instance overlapping problem to improve the transformed video's utility on video analytical tasks. Gaussian filter is applied to auxiliary features to prevent potential privacy leakage. Using different datasets, we evaluate our system's privacy protection guarantees and the utility of the transformed videos. We also give strategies for selecting an auxiliary feature and applying the Gaussian filter. Extensive evaluation results show the superior performance of our system compared to existing privacy-preserving video transformation solutions.

References

[1] D. I. M. Size and D. I. M. S. Growth, "Share & trends analysis report by product (titanium implants, zirconium implants), by region (north america, europe, asia pacific, latin america, mea), and segment forecasts, 2018-2024," *Personalized Medicine Market Analysis By Product And Segment Forecasts To 2022*, 2018, (accessed date: 03/11/2023).

[2] "Dash Camera Tops List of Features Wanted by Future Vehicle Buyers," https://www.autopacific.com/autopacific-insights/2020/7/17/dash-camera-tops-list-of-features-wanted-by-future-vehicle-buyers, (accessed date: 03/11/2023).

[3] A. Frome, G. Cheung, A. Abdulkader, M. Zennaro, Bo Wu, A. Bissacco, H. Adam, H. Neven, and L. Vincent, "Large-scale privacy protection in Google Street View," in *IEEE ICCV*, 2009, pp. 2373–2380.

[4] S. Liu, L. Liu, J. Tang, B. Yu, Y. Wang, and W. Shi, "Edge Computing for Autonomous Driving: Opportunities and Challenges," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1697–1716, 2019.

[5] C. Bloom, J. Tan, J. Ramjohn, and L. Bauer, "Self-driving cars and data collection: Privacy perceptions of networked autonomous vehicles," in *USENIX SOUPS*, 2017, pp. 357–375.

[6] Y. Zhao, Y. Yao, J. Fu, and N. Zhou, "If sighted people know, i should be able to know: Privacy perceptions of bystanders with visual impairments around camera-based technology," *arXiv:2210.12232*, 2022.

[7] G. Kapteinis, "Vehicle mounted dashboard cameras: a practical approach to gdpr compliance," 2021.

[8] L. C. Klopfenstein, S. Delpriori, P. Polidori, A. Sergiacomi, M. Marcozzi, D. Boardman, P. Parfitt, and A. Bogliolo, "Mobile crowdsensing for road sustainability: Exploitability of publicly-sourced data," *International Review of Applied Economics*, vol. 34, no. 5, pp. 650–671, 2020.

[9] H. Kaur and J. Sahambi, "Vehicle Tracking in Video using Fractional Feedback Kalman Filter," *IEEE Transactions on Computational Imaging*, pp. 1–1, 2016.

[10] A. Ottlik and H.-H. Nagel, "Initialization of Model-Based Vehicle Tracking in Video Sequences of Inner-City Intersections," *International Journal of Computer Vision*, vol. 80, no. 2, pp. 211–225, 2008.

[11] Jianpeng Zhou and Jack Hoang, "Real Time Robust Human Detection and Tracking System," in *IEEE CVPR*, vol. 3, 2005, pp. 149–149.

[12] J. Gao, A. Kosaka, and A. C. Kak, "A multi-Kalman filtering approach for video tracking of human-delineated objects in cluttered environments," *Computer Vision and Image Understanding*, vol. 99, no. 1, pp. 1–57, 2005.

[13] Z. Tang, M. Naphade, S. Birchfield, J. Tremblay, W. Hodge, R. Kumar, S. Wang, and X. Yang, "PAMTRI: Pose-Aware Multi-Task Learning for Vehicle Re-Identification Using Highly Randomized Synthetic Data," in *IEEE ICCV*, 2019, pp. 211–220.

[14] X. Liu, W. Liu, T. Mei, and H. Ma, "A Deep Learning-Based Approach to Progressive Vehicle Re-identification for Urban Surveillance," in *IEEE ECCV*, 2016, vol. 9906, pp. 869–884.

[15] B. He, J. Li, Y. Zhao, and Y. Tian, "Part-Regularized Near-Duplicate Vehicle Re-Identification," in *IEEE CVPR*, 2019, pp. 3992–4000.

[16] K. Zhou, Y. Yang, A. Cavallaro, and T. Xiang, "Omni-Scale Feature Learning for Person Re-Identification," in *IEEE ICCV*, 2019.

[17] W. Li, R. Zhao, T. Xiao, and X. Wang, "DeepReID: Deep Filter Pairing Neural Network for Person Re-identification," in *IEEE CVPR*, 2014, pp. 152–159.

[18] M. Hirzer, C. Beleznai, P. M. Roth, and H. Bischof, "Person Re-identification by Descriptive and Discriminative Classification," in *Image Analysis*, 2011, vol. 6688, pp. 91–102.

[19] X. Liu, W. Liu, T. Mei, and H. Ma, "PROVID: Progressive and Multimodal Vehicle Reidentification for Large-Scale Urban Surveillance," *IEEE Transactions on Multimedia*, vol. 20, no. 3, pp. 645–658, 2018.

[20] H. Wu, J. Feng, X. Tian, F. Xu, Y. Liu, X. Wang, and S. Zhong, "secGAN: A Cycle-Consistent GAN for Securely-Recoverable Video Transformation," in *HotEdgeVideo*, 2019, pp. 33–38.

[21] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft COCO: Common Objects in Context," *arXiv:1405.0312*, 2015.

[22] J. Yu, H. Xue, B. Liu, Y. Wang, S. Zhu, and M. Ding, "GAN-Based Differential Private Image Privacy Protection Framework for the Internet of Multimedia Things," *Sensors*, vol. 21, no. 1, p. 58, 2020.

[23] J. Fan, H. Luo, M.-S. Hacid, and E. Bertino, "A novel approach for privacy-preserving video sharing," in *ACM CIKM*, 2005, pp. 609–616.

[24] R. Uittenbogaard, C. Sebastian, J. Vijverberg, B. Boom, D. M. Gavrila, and P. H. de With, "Privacy Protection in Street-View Panoramas Using Depth and Multi-View Imagery," in *IEEE CVPR*, 2019, pp. 10 573–10 582.

[25] A. Nodari, M. Vanetti, and I. Gallo, "Digital privacy: Replacing pedestrians from Google Street View images," in *IEEE ICPR*, 2012, p. 5.

[26] H. Wu, X. Tian, M. Li, Y. Liu, G. Ananthanarayanan, F. Xu, and S. Zhong, "PECAM: Privacy-enhanced video streaming and analytics via securely-reversible transformation," in *Mobicom*, 2021, pp. 229–241.

[27] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *arXiv:1311.2524*, 2014.

[28] R. Girshick, "Fast R-CNN," *arXiv:1504.08083*, 2015.

[29] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *IEEE CVPR*, 2016, pp. 779–788.

[30] G. Jocher, A. Chaurasia, A. Stoken, J. Borovec, NanoCode012, Y. Kwon, K. Michael, TaoXie, J. Fang, imyhxy, Lorna, Z. Yifu, C. Wong, A. V, D. Montes, Z. Wang, C. Fati, J. Nadar, Laughing, UnglvKitDe, V. Sonck, tkianai, yxNONG, P. Skalski, A. Hogan, D. Nair, M. Strobel, and M. Jain, "ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation," Nov. 2022. URL: https://doi.org/10.5281/zenodo.7347926 (accessed date: 03/11/2023).

[31] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *Medical Image Computing and Computer-Assisted Intervention*, 2015, vol. 9351, pp. 234–241.

[32] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking Atrous Convolution for Semantic Image Segmentation," *arXiv:1706.05587*, 2017.

[33] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *NeurIPS*, vol. 27, 2014.

[34] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network," *arXiv:1609.04802*, 2017.

[35] T. Karras, S. Laine, and T. Aila, "A Style-Based Generator Architecture for Generative Adversarial Networks," *arXiv:1812.04948*, 2019.

[36] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-Image Translation with Conditional Adversarial Networks," *arXiv:1611.07004*, 2018.

[37] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional gans," in *IEEE CVPR*, 2018.

[38] 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," in *IEEE CVPR*, 2016, pp. 3213–3223.

[39] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, 1986.

[40] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved Training of Wasserstein GANs," *arXiv:1704.00028*, 2017.

[41] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian, "Scalable Person Re-identification: A Benchmark," in *IEEE ICCV*, 2015.

[42] E. Ristani, F. Solera, R. S. Zou, R. Cucchiara, and C. Tomasi, "Performance measures and a data set for multi-target, multi-camera tracking," in *ECCV Workshops*, 2016.

[43] A. Kumar, T. Kashiyama, H. Maeda, F. Zhang, H. Omata, and Y. Sekimoto, "Vehicle re-identification and trajectory reconstruction using multiple moving cameras in the carla driving simulator," *IEEE Big Data*, pp. 1858–1865, 2022.

[44] M. Broström, "Real-time multi-camera multi-object tracker using YOLOv5 and StrongSORT with OSNet," 2022. URL: https://github.com/mikel-brostrom/yolov8_tracking (accessed date: 03/11/2023).

[45] tfaehse, "Tfaehse/DashcamCleaner," 2022. URL: https://github.com/tfaehse/DashcamCleaner (accessed date: 03/11/2023).

[46] Q. Guo, W. Feng, R. Gao, Y. Liu, and S. Wang, "Exploring the effects of blur and deblurring to visual object tracking," *IEEE Transactions on Image Processing*, vol. 30, pp. 1812–1824, 2021.

[47] "Mrnuclear8/SecStudent," https://github.com/mrnuclear8/SecStudent, 2021, (accessed date: 03/11/2023).

[48] Z. Zheng, X. Yang, Z. Yu, L. Zheng, Y. Yang, and J. Kautz, "Joint discriminative and generative learning for person re-identification," *IEEE CVPR*, 2019.

[49] Z. Zheng, M. Jiang, Z. Wang, J. Wang, Z. Bai, X. Zhang, X. Yu, X. Tan, Y. Yang, S. Wen *et al.*, "Going beyond real data: A robust visual representation for vehicle re-identification," in *IEEE CVPR Workshops*, 2020, pp. 598–599.

[50] Z. Zheng, T. Ruan, Y. Wei, Y. Yang, and T. Mei, "Vehiclenet: Learning robust visual representation for vehicle re-identification," *IEEE Transactions on Multimedia*, vol. 23, p. 2683–2693, 2021.

[51] 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," *arXiv:2106.12083*, 2018.

[52] F.-H. Chan, Y.-T. Chen, Y. Xiang, and M. Sun, "Anticipating accidents in dashcam videos," in *AFCV ACCV*, 2017, pp. 136–153.