Privacy-Preserving Analysis for Remote Video Anomaly Detection in Real Life Environments

Giacomo Giorgi*, Wisam Abbasi, and Andrea Saracino

Institute for Informatics and Telematics, National Research Council of Italy
{giacomo.giorgi, wesam.alabbasi, andrea.saracino}@iit.cnr.it

Received: December 19, 2021; Accepted: February 10, 2022; Published: March 31, 2022

Abstract

This paper proposes a novel approach for privacy-preserving surveillance video streams anomaly detection, i.e., situations implying violence, illegal actions, or situations involving hazards. In particular, this approach adopts a privacy-preserving mechanism based on autoencoder neural networks applied in a differential private manner, exploiting three different types of differential private optimizers. Recorded real-world video streams are segmented into data frames, which are compressed into special codes with autoencoders and differential privacy and transmitted to a central server where they get decoded into an anonymized version of the original data frame that can be analyzed to detect anomalies. The anomaly detection algorithm exploits a supervised learning binary classification methodology of extracted contextual, spatial, and motion data on imbalanced datasets. Anomalies are differentiated into ”soft” and ”hard”, and the anomaly detection score is computed based on a sigmoidal function. The proposed methodology has been validated with a set of experiments on a well-known video anomaly dataset: UCF-CRIME. The experiments we conducted on the testbed demonstrate the capability of the system to correctly identify video anomalies, with a consistent privacy gain demonstrated by the strongly reduced ability to identify people from faces in the reconstructed frames.

Keywords: Anomaly detection, Autoencoders, Behavioral analysis, Deep Learning, Computer vision, Differential Privacy, Trustworthy Artificial Intelligence.

1 Introduction

Currently, a large amount of video information is daily generated by new generation ubiquitous devices like smartphones, webcams, surveillance cameras, etc. These devices can capture a real-world action in every single moment of the day. Furthermore, the evolution of the Internet of Things (IoT), and the rapid growth of smart cities have led to a large diffusion of surveillance cameras, also known as Closed-Circuit Television (CCTV). These cameras can be connected to the network, and the analysis of what they can capture would be extremely beneficial in implementing real-time and real-world monitoring and surveillance applications. For instance, person identification to identify those who are accessing to a restricted area (companies, airport, buildings) [1, 2], person/vehicle tracking to follow the movement of a specific target [3], Human action recognition to detect prohibited action in a controlled environment [4], or anomaly detection to identify events or activities that are unusual and prohibited in a surveillance environment [5]. The automation of video anomaly detection is one of the main tasks in the video

Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, 13(1):112-136, Mar. 2022
DOI: *10.22667/JOWUA.2022.03.31.112*

*Corresponding author: Institute for Informatics and Telematics, National Research Council of Italy, Via G. Moruzzi, 1, 56124, Pisa, Italy surveillance field. Such an automation, by monitoring the video camera stream, should aim at detecting anomalous behavior as quickly as possible, to enable countermeasure for mitigating the anomaly. However, automating the anomaly detection process is challenging due to noise factors that can impact the scene, e.g., different perspective views, human pose, lighting variability, and occlusions. In addition, the detection systems exploits high computational capabilities, and very often such analysis is demanded to a third-party remote server endowed of costly and dedicated hardware.

However, the continuous monitoring of a public or private environment might lead to severe *privacy concerns* due to the sensitive information involved in videos such as human faces, objects, identities, or human activities. Furthermore, in the case of remote analysis, the transfer and the managing of video stream containing sensitive information is a risky process for *data confidentiality* and *data breach*. Depending from the specific application field, this might be in violation of the EU proposed guidelines for the usage of artificial intelligence[1], and GDPR in general. Moreover, personal data could be the target of cyber-attacks. To this end, video sanitization and anonymization techniques have been developed in recent years. Object/faces removal techniques or frame blurring and noise addition mechanisms are used to maintain privacy [6]. However, such alterations reduce accuracy (utility loss reduction) in the anomaly detection phase. Hence, a depth study on the trade-off between privacy and accuracy is relevant.

In our work, we propose a remote video anomaly detection framework that guarantees privacy, by adding noise to the video frames to impair the recognizing of sensitive information, and ensure data confidentiality during the communication with the remote data analysis server. At the same time, we demonstrate how our system is still able to recognize anomalous behaviors. The conducted experiments focus on analyzing the trade-off between privacy and the utility loss.

The contributions of this work are the following:

- We present a privacy preserving mechanism based on autoencoder technique applied to preserve privacy and confidentiality.
- We evaluate different privacy preserving training techniques performed on the autoencoder.
- We present the implementation of a remote video anomaly detection applied in a smart environment scenario, which exploit the proposed framework to ensure privacy during the video analysis performed by a third-party service provider.
- We evaluate the privacy mechanism on frames quality based on the applied privacy degree.

This paper is an extension of the work presented in [7], which presents as new contributions:

- the proposal of a privacy preserving technique for video streams anomaly detection based on Autoencoder Neural Networks and Differential Privacy.
- The application of the framework on a use case scenario in which the data analysis is performed on a remote server preserving data privacy and confidentiality.
- An evaluation of the privacy mechanism on the anomaly detection accuracy.

The rest of the paper is organized as follows: Section 2 lists some related works for video streams anomaly detection and privacy preserving approaches in computer vision field. Section 3 describes the used architecture in addition to the design and implementation of the proposed methodology. Section 4 reports the used dataset and experimental setup and implementation. Section 5 reports the experiments results and evaluate the proposed model performance in terms of accuracy and privacy. Finally, Section 6 briefly concludes the paper and proposes some future directions.

[1] https://eur-lex.europa.eu/legal-content/EN/TXT/?qid=1623335154975&uri=CELEX%3A52021PC0206

2 Related Works

This section lists the related work for anomaly detection mechanisms used in video analytics and privacy preserving techniques in computes vision field.

2.1 Video Anomaly Detection

Video anomaly detection is one of the most complexes and studied problems of computer vision. An *anomaly* is any pattern that does not conform to what is considered *expected* [2]. The two classes that characterize the problem are defined as one the negation of the other and both can take completely different forms, specific to the problem under consideration, i.e., a person who runs can be considered an anomaly within a public office but is expected in contexts such as stations or parks. Moreover, it is not easy to establish a priori all the possible anomalies that may occur even considering a single context.

Handcrafted Methods. The first methods of anomaly detection were *trajectory-based* [8, 9]. The main idea is to identify the distinct trajectories of objects within *normal videos* of expected behaviour with no anomalies. The anomalies are highlighted as objects that do not follow similar trajectories. However, these methods are of restricted applicability and can be used only in the presence of constant and unobstructed trajectories. The use of other handcrafted features allows enriching the ability of a detector to identify more general classes of anomalies, not limited to trajectories only. These low-level features generally extrapolate information about appearance, movement, and texture. *Histograms of optical flows* [10], *histograms of oriented gradients* [11], *social forces maps* [12] and *mixture of dynamic textures* [13], are just some of the methods developed. Although very effective in identifying specific anomalies, these feature extraction methods cannot adapt to categories of abnormalities not previously seen.

Semi-Supervised Methods. To overcome these problems, some of the most used approaches are typically *semi-supervised*. This learning method category uses only normal videos to train the detector to identify anomalies as any deviation from the notion of normality that they have learned. Precisely avoiding giving a specific characterization to anomalies allows this type of detector to be more robust towards types of abnormalities not initially foreseen. Another great help given to the generalization capabilities of anomaly detectors is the use of features extracted through *Deep Neural Networks* (DNNs) [14]. Neural networks allow to autonomously extract semantically significant features that can introduce a better generalization capability with respect to the handcrafted ones. The current state-of-the-art combines semi-supervised approaches and neural networks. Among the most popular approaches in literature, we can mention *autoencoders* [15] and *Generative Adversarial Networks* (GAN) [16]. These networks are usually trained on images – frame and optical flow – extracted from normal videos to reconstruct them or predict the next in time order [17]. When anomalous images are presented to the network this is generally not able to recreate them, as it is trained only on normal images. The anomaly generates a greater reconstruction error, allowing it to be distinguished. Although certainly more robust than handcrafted methods, techniques based on image reconstruction also have limitations, i.e., the networks may be able to reconstruct even the anomalies [18], not allowing them to be distinguished. In [19] it is highlighted that deep learning method are also characterized by a lack of explainability. In this work, the GradCam tool [20] is used as a method to locate the regions in a frame that contributes most to the assignment of a higher reconstruction error in their auto-encoder based approach.

[2] Anomalous and normal in this paper are used as strictly technical terms in the context of intrusion/danger detection and are not intended to imply any discrimination.

Supervised Methods. These approaches involve the use of labeled videos to reduce the problem of binary classification. The main obstacle to studying these methods is the non-availability of large labeled datasets, which are very expensive to produce. Recently [5] introduced a new dataset for the detection of real-life anomalies concerning crimes – e.g., robberies, assaults, shootings – with more than 128 hours of untrimmed videos (*UCF-Crime*). The same paper proposes a *multiple instance learning* method that allows the training of *weakly supervised* binary classifiers using labels at video level. More recently in [21] the UCF-Crime dataset has been enriched with spatiotemporal annotations (*UCFCcrime2local*), allowing the experimentation of *strongly supervised* methods. Some commonly used neural networks in a supervised environment are *3D convolutional networks* [22]. These are also often exploited in the field of *action recognition* [23] and allow to extrapolate spatiotemporal features able to describe the actions inside video segments. They can also be used in the form of *two-stream networks* [24] to extract features from streams with different frame rates or using an optical flow stream in parallel with the video stream. However, 3D networks have the problem of being particularly heavy, both for training and inference. For practical applications, lighter networks such as *2D-CNNs* can be taken into consideration by enriching their capabilities using solutions to add time and motion information to the spatial features extracted from the network. Temporal information can be added, for example, by feeding *Long-Short-Term-Memory networks* (LSTM) with the spatial features extracted by the CNN as in [25] while motion information can be included using a two-stream solution [26].

2.2 Privacy Preserving Video Analytics

With the widespread deployment of surveillance cameras and great advances in video analytics and computer vision fields, privacy and security have become major concerns raising social, ethical, and legal issues. Therefore, adequate implementation of privacy preserving mechanisms has been an active area of research during the last years to help mitigate these concerns. Privacy preserving mechanisms are used to hide sensitive and identifying attributes in the datasets that might reveal the identity of a person or other critical information. Several privacy methods has been adopted in the computer vision field such as video anonymization mechanisms by reducing the resolution quality of the video [27] or using image obfuscation operations in what is called denaturing, which involves modifying the original content of an image or video frame to hide sensitive attributes [28, 29]. These operations include image blurring [30, 31, 32], pixelating [33, 34, 35], cartoon effects addition [36], face swapping of person in the image with a similar pose [37], and face de-identification methods that alter the faces in an image to protect the person identity such as the K-Same algorithm [38] and its extention, the K-Same-Select algorithm [39]. In addition to the methods used to remove people and objects from images and videos [40], followed by the use of inpainting techniques to repair the missing parts of an image or a frame after reconstruction [41, 42, 43].

The issue in such approaches, that they completely remove the original face or object, obfuscate the image in a manner that causes a major drop in accuracy, or do not provide strong privacy guarantees. Thus, a mechanism that provides strong provides strong privacy guarantees while preserving a balance with the models accuracy is required. *Autoencoders* [44], a kind of neural networks have also been used for privacy preserving purposes like in [45, 46, 47]. Autoencoders are used to compress data into a special code holding the most representative features of the original input data and then reconstruct an altered version of the input data using this code to protect individuals privacy, Since the reconstructed data is anonymized due to the loss of some attributes, and the degree of privacy is controlled by the code size. *Differential Privacy (DP)*[48], is a powerful mechanism that has been used for privacy preserving by adding noise to the data either before training phase using Synthetic dataset generation approaches [49, 50], or during training phase by adding noise to the gradients [51].

3 Proposed Methodology

This section presents in details the proposed model, the techniques used for private data protection, the techniques for efficient and effective data utilization during training with a novel approach for anomaly score.

3.1 Reference Scenario

In our scenario we considered an anomaly detection service on the cloud which provides remote analysis for detecting anomalies in a customer video stream. Cloud services are becoming more and more popular because they offer the possibility to exploit a service that it can be executed on a server with high computational capabilities. However, sending sensitive data across the network is a risky operation. To preserve data confidentiality and integrity, several protocol have been proposed such as TLS, SSL that are able to encrypt the communication and ensure data confidentiality. however, these protocols are not free from possible attacks [52]. Another aspect to be taken in consideration is the treatment of sensitive data by the remote service. Sending sensitive data to a remote server needed a trust agreement between the customer and the service provider. Several trust model have been theorized from those that do not guarantee any level of privacy, i.e., the *full-trust* model, where the service provider has full access to all of the data of its customers and is trusted not to abuse its privileges. Till those that guarantee the greatest degree of privacy but do not provide any margin for data analysis, i.e., *zero-trust* model, where the service provider holds but cannot gain access to the decrypted data at the servers and has limited or no insights about the data it holds. Such trade-off between privacy and data analysis space is a challenging task on which is focused our framework. Figure 1 depicts the working schema of the proposed framework. The reference scenario is implemented in a surveillance environment (a) composed of a smart camera with low computational capabilities. The camera is used to monitor the scene that is taking place in the surveillance environment. Due to the smart camera's low computational capabilities, part of the computational intelligence is delegated to a server (h) endowed with high capabilities (GPUs, memory, storage). The smart camera captures the scene in the surveillance environment and apply a preprocessing step before sending the video frames to the server to guarantee confidentiality and reduce the size of the data transmitted: each clear frame (b) captured from the scene is passed to a compression network (deployed in the smart camera) implemented with an encoder (c). The entire autoencoder is trained externally and the encoder part is deployed in the smart camera while the decoder part is deployed on the server. The output of the encoder is a representation of the compressed input frame, called latent space (d) and it is send through a secure connection (TLS communication) to the server. The encoder acts as a compressor to reduce the space dimensionality, removing all the redundant information and only focusing on the most important features. In addition, the encoder is trained with a differential privacy strategy that guarantees privacy, mitigating the risk of exposing sensitive training data in the synthetic data model or its output. The fact that the communication between client and server happens transferring only the latent space information provides *confidentiality* to the client data and reduces the communication overhead. Even if the communication is intercepted, the extraction of information from the obtained latent space is inherently infeasible. On the server side, the latent space received is passed to the decoder part (e), which is able to reconstruct a similar version of the original frame. The compression performed by the encoder on the client-side produces a frame resolution decay in the reconstruction frame (f) that guarantees *anonymization* of the captured scene. Finally, the anomaly detection network (g) analyses the reconstructed video and computes an anomaly score.

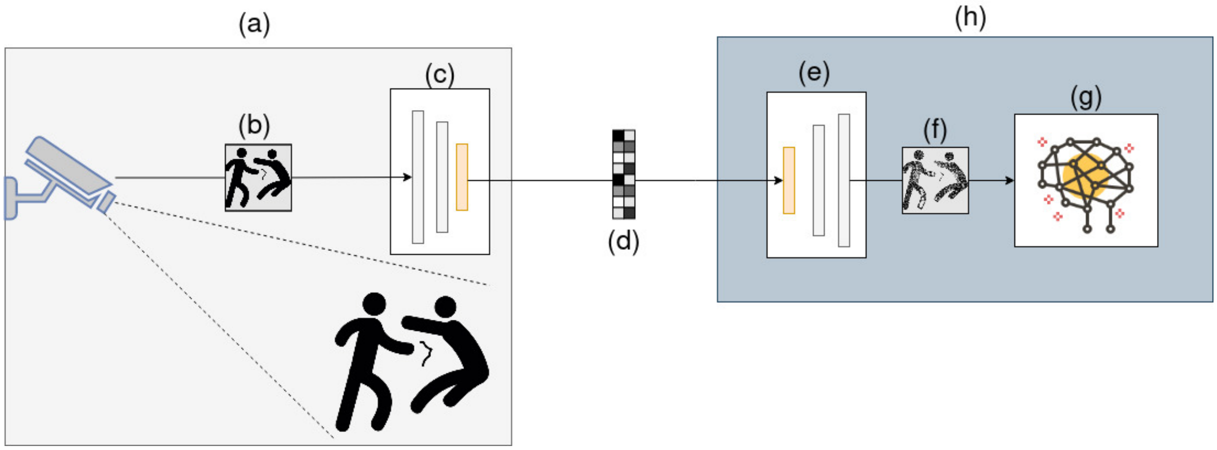


Figure 1: Privacy preserving video anomaly detection scenario

3.2 Architecture

The proposed architecture is presented in Figure 2. The model starts with an autoencoder neural network divided in two parts, the first part lays at the client side as an encoder to encode in a differential private manner the captured frame into a latent space representation, which is a compressed representation of the original image with a reduced dimension. The produced latent representation is forwarded to the server side to be decoded using the second part of the autoencoder: the decoder. The decoder reconstructs the original frame but with a lower data utility. The reconstructed frames are then used to calculate the optical flow between current and previous reconstructed frames. The reconstructed frames with the optical flow are then passed to the double stream *ResNet-50* network [53] to extract their features. Additionally, A *YOLOv4* [54] object detector uses the current reconstructed frame to generate a vector of identified objects per class in the frame with their count, which is referred to as *bag-of-objects*. As a final step, the features extracted from the double stream network are concatenated with the bag-of-objects and forwarded to the fully connected layers of the network to produce a final anomaly score for the analyzed video. All concepts exploited in this architecture are detailed in the following subsections.

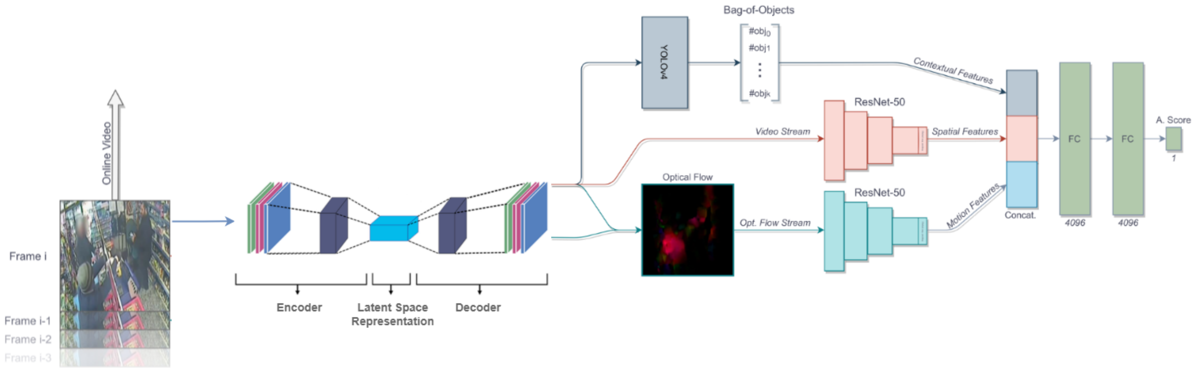


Figure 2: Privacy Preserving Anomaly Detection Architecture

3.3 Privacy Preserving Mechanisms Enforcement

Data privacy is preserved by means of Autoencoders and Differential privacy techniques. These techniques do not remove faces and objects from the original frame and their degree of privacy can be controlled in a manner that keeps a balance with the anomaly detection model accuracy. Autoencoders are used to alter the original frame through compression and reconstruction operations, which result in frame disturbing and some utility loss. Combined with differential privacy embedded in autoencoders as an optimization mechanism, the model memorization of specific instances and dataset attributes is restricted, and only general data patterns are learned. Both methods will be detailed in the following subsections.

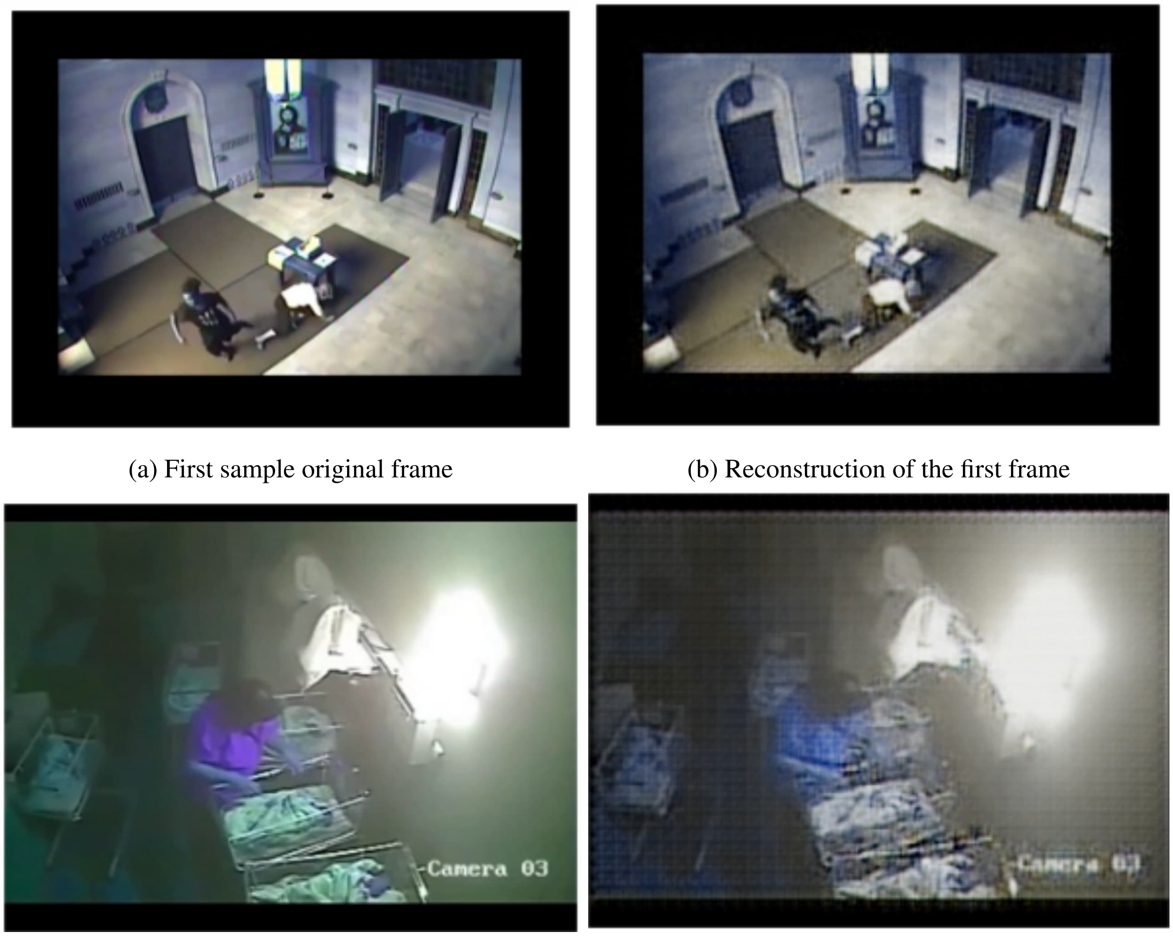
3.3.1 Autoencoders

To preserve the privacy of data collected as video streams, frames are passed through autoencoder neural networks. Autoencoders are a data compression mechanism [47, 55] used for dimensionality reduction and feature representation as a latent space [56]. This type of neural networks consists of two parts, the first is responsible for latent space representation construction using the original input image and is referred to as encoder. The second part is responsible for the reconstruction of the original image from the latent space and is defined as the decoder. Autoencoders generally results in a loss of utility for the reconstructed image compared to the original image due to the loss of some features during the compression process. As illustrated in Fig 3, where sub-figures (a) and (c) represents original frames collected from a video stream at the client side and sub-figures (b) and (d) represents reconstructed frames at the server side. This loss of utility is measured by the distance between the input frame image and the reconstructed frame image. Thus, autoencoder networks are used as a privacy preserving mechanism [45, 57, 58] to protect sensitive data within a dataset. Aautoencoders are composed of an encoder, which is used as a compression algorithm for dimensionality reduction and a decoder part to recreate the original input. The result of the encoder produces a latent space image that represents encryption of the input data. For this reason, even if a third party acquires such data, it should not be able to decrypt it without the knowledge of the decoder part. Such a mechanism can be considered a system to reach confidentiality during the transmission of data between a client (endowed by the encoder) and server (endowed by the decoder).

Implementation details The architecture of the autoencoder is composed by two subnetwork. The *encoder* consists of an Input Layer of size equal to the RGB frame dimension 240 *x* 320 *x* 3. Three convolutional layers are sequentially applied to the input data having respectively 240, 64 and 16 filters with kernel size 3×3 and strides 2. Their aim is to reduce the dimensionality of the input space producing a latent space representation having size $30 \times 40 \times 16$. The decoder part, increasing the dimensionality of the output of the encoder through three transposed convolutional layers, whose reverse the operation of the standard convolutional layers (deconvolution operation), and a final convolutional layer to obtain the reconstructed input frame. The overall autoencoder network schema is represented in figure 4.

3.3.2 Differential Privacy

A state-of-the-art and one of the most powerful privacy-preserving mechanisms is differential privacy, which is used to add noise either to the data before the analysis phase or during data analysis based on Laplace or Gaussian distributions in a way that makes individual data instances indistinguishable when added to or removed from a dataset. This privacy mechanism uses privacy and sensitivity parameters to



(a) First sample original frame

(b) Reconstruction of the first frame

(c) Second sample original frame

(d) Reconstruction of the second frame

Figure 3: Samples of original frames collected from a video stream and their reconstructed frames using an autoencoder implementation

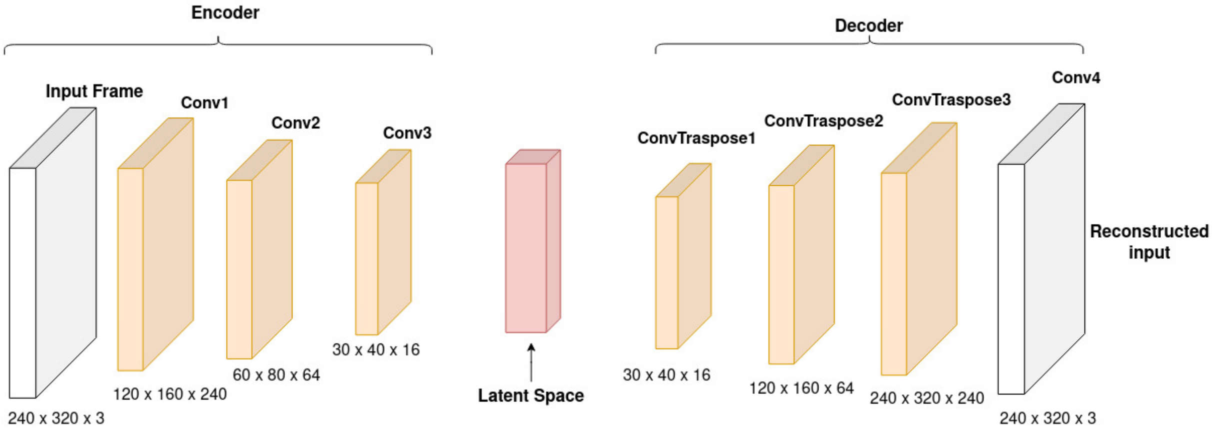


Figure 4: Autoencoder network

control the privacy degree applied to the dataset using Equation (1) [48].

$$Pr[K(D_1) \in S] \leq Pr[K(D_2) \in S] \times exp(\varepsilon) + \alpha \tag{1}$$

Where ε represents the privacy budget, α is the probability of failure, K is the randomized function providing ε-differential privacy for both datasets D_1 and D_2 which differ in one element only, and $S \subseteq Range(K)$.

In our proposed approach, Tensorflow-Privacy framework [3] is used to add noise to the to the gradients at each iteration of the training phase of the autoencoder neural network using a differential private optimizer, so that the datasets elements are encoded and decoded by the model without model memorization or leakage of a specific element sensitive features.

3.4 Video Anomaly Detection

Anomaly detection of video streams is performed using information of different types and collected at multiple levels. This section presents the anomaly detection methodology and mechanisms exploited in our approach.

3.4.1 Contextual Information Extraction

contextual features are extracted using a state-of-the-art version of YOLO object detector is *YOLOv4* [54], which is an enhanced version in terms of accuracy, performance, average precision, and real-time speed measured by frames count per second allowing for real-time video analysis. YOLOv4 has been pre-trained on a large-scale object detection dataset referred to as *MS COCO* [59]. This dataset contains 80 object categories of people, food, animals, vehicles, and things. Reconstructed video frames using autoencoders are passed to YOLO to extract contextual features represented by the bag-of-objects explained in 3.2. These classes of objects in specific contexts such as that of UCF-Crime, might be of importance for the classifier downstream in order to weigh these features against other factors such as temporal and spatial information.

3.4.2 Spatial and Motion Information Extraction

Spacial features represent the locations of the identified objects within a frame image, and motion features are extracted to identify patterns of apparent motion caused by a relative movement in a video stream. These features, spacial and motion features, are extracted by a two-stream architecture presented in Figure 2, which involves a pre-trained *ResNet-50* [53] on *ImageNet* [60] as base convolutional network. This two-stream architecture takes reconstructed frames and the optical flow generated using reconstructed frames from the video that is being analyzed as two inputs, where the optical flow represents a pattern of apparent motion caused by the relative movement between an observer and a scene [61]. A convolutional layer is used to extract spatial and motion features from the two-streams using a global average pooling. Then, these features are being concatenated with the contextual information extracted at the previous step through a *joint fusion* architecture. As a final step, the concatenated result is forwarded as an input to a fully connected classifier of two fully connected layers with *ReLu* activation, consisting of 4096 neurons each and connected to a final neuron that produces as an anomaly score the *anomaly class confidence* between $[0,1]$.

[3] https://github.com/tensorflow/privacy

3.4.3 Data Balancing Strategies

Imbalanced datasets classification is a major issue when training a classification model such as an anomaly detection model, where the distribution of dataset elements across classes is skewed and imbalanced. In our case, it is concerned with the count of video frames considered as normal compared to the anomalous frames and their distribution in the video being analyzed. To solve this issue, a training dataset of videos with as many possible classes of anomalies need to be used and different scenarios and contexts also need to be considered for each type of anomalies. To do this, we are considering different sampling frame-rates for anomalous and normal segments and hard mining in our approach as explained below:

Adaptive Frame-rate Sampling: to solve the imbalanced dataset and its classification problem, it is necessary to subdivide videos into frames or adjacent frame segments for 2D-ConvnNets and 3D-ConvNets respectively according to the pre-defined frame-rate, which is defined as Frames Per Second (FPS) such as 30 FPS for UCF-Crime dataset. The issue of applying this method is that it generates a great amount of adjacent frames with slight differences. Thus increasing redundant data to be analyzed and the analysis cost in addition to model overfitting problems, in which the training model learns specific details about the training data and bias in a way that affects model performance negatively. In order to overcome these limitations, a lower frame-rate can be used with the same resources providing the opportunity to use a wide variety of videos with various anomalies and contexts for better generalization of the anomalous and expected patterns, thus better pattern recognition. A further enhancement to this mechanism would be the adaptive frame-rate sampling implementation, where an even lower frame-rate could be used with normal videos in such a way that the undersampling does not effect the quality of the training process. This method is also applicable to 3D-CNNs by selecting a higher stride value for anomalous video segments than normal segments during training phase, where stride represents the overlap between two segments extracted consecutively from a video stream.

Hard Sample Mining: this method has been proposed to address the issue of very large scale and extremely imbalanced training datasets, it involves using batch-wise incremental hard sample mining of minority attribute classes, by selecting selecting samples with greater loss to be used during the training phase based on the Class Rectification Loss(CRL) regularising algorithm [62]. In our approach, we used batches of K samples during training and the sampling is performed without replacement αK candidates frames, where α is a multiplicative factor for each patch. The steps followed are as below:

1. Loss calculation using an inference operation.

2. Only first k samples with higher loss are selected despite whether they are hard-positives or hard-negatives.

3. The k samples are then used as areal batch training, where only $\frac{1}{\alpha}$ of the total training samples are used.

Anomaly detection problems are binary class problems of two output classes: Normal and Anomaly. Normal class represent expected patterns. Even though, anomaly class involves several sub-classes for anomalies, which have equal importance and require a balanced distribution of classes within the dataset and to have various samples of each class in the dataset in order to be recognized accurately by the analysis model. Thus, hard sample mining is useful for adaptive sampling of these sub-classes in each epoch.

3.4.4 Anomaly Types

For videos containing anomalous behavior, this behavior occurs in some specific segments. The distinction between these segments and other segments that come right before or after the anomalous action might become difficult. Considering these other segments, unusual action can be guessable either before or after its occurrence. For instance, a car robbery or stealing objects from a car might be guessed from suspicious actions happening before the theft, like a stranger looking through the windows of the vehicle or a broken glass after the occurrence of the theft. These segments are considered *soft anomalies*, as they have suspicious behaviour but not a real anomalies (*hard anomalies*), like the segments of the theft or the crime itself. Thus, to improve the training quality, soft anomalies labeling as fully normal or fully anomalous must be avoided.

3.4.5 Anomaly Scoring Strategy

Anomaly score computation is preferred to a noise-robust mechanism, rather than performing direct online anomaly detection based on the output result of the classification model [63]. For instance, using the confidence value of the anomalous classes as an anomaly scoring strategy with a pre-defined alarming threshold is possible, still it is not a noise-robust method and would result in recurring false positives. Thus, a ***sigmoidal anomaly score*** (SAS) mechanism has been proposed and used in our approach for noise-robust anomaly score computation using two parameters: *sensibility* and *reactivity*. The anomaly score s_t is calculated as:

$$s_t = S(x_t) = \frac{1}{1 + e^{-x_t}}$$

Where $S : R \to [0, 1]$ is a standard logistic sigmoid function and x_t the value of the *accumulator* at time t computed as:

$$x_t = \left\{ \begin{array}{l} x_{t-1} + \Delta_t^+ \nu\&\text{if } \sigma_t \geq \tau \\ x_{t-1} - \Delta_t^- \nu\&\text{if } \sigma_t < \tau \end{array} \right. \quad with\, x_t \in [LB, UB]$$

Where σ_t is the anomaly class confidence calculated at time t by the classifier, x_{t-1} is the accumulator's value calculated at previous step, $\tau \in [0, 1)$ is the sensibility threshold and $\nu \in (0, +\infty)$ the reactivity parameter. $\Delta_t^+, \Delta_t^- \in [0, 1]$ can be determined as:

$$\Delta_t^+ = \frac{\sigma_t - \tau}{1 - \tau}; \Delta_t^- = \frac{1 - \sigma_t - \tau}{1 - \tau}$$

Which represent the increase and decrease in anomaly score due to going beyond or below sensibility threshold τ. The closer τ gets to 1, the more confidence is needed in anomaly classification to increase the score value. And the closer τ is to 0, the more sensitive the score is to lower confidence classifications. Whereas, reactivity ν is a multiplicative factor which allows to decide how quickly the score increases or decreases. A higher value of ν increases the sensitivity to anomalies taking place in short time intervals, but increase the possibility to have more false positives due to the higher possibility of having noise. In contrast, a value below 1 for ν allows to filter out noise in a more effective manner, making the score less sensitive to sudden anomalies. To avoid that the accumulator x assumes too high or low values, these are limited by a lower and upper bound (LB, UB).

4 Experiments

This section reports the experiments performed to validate the proposed approach. It provides a detailed description of the dataset used for training anomaly detection and autoencoder. Subsequently are described in detail the experiments conducted on the privacy preserving and anomaly detection.

4.1 Dataset

The performed experiments are based on the public reference dataset for anomaly detection *UCF-Crime*. From this we formulated two different dataset used respectively for anomaly detection training and autoencoder training. The details are reported in the following sections.

4.1.1 Anomaly detection dataset

In this study, we considered the public UCF-Crime dataset [5]. This dataset consists of 128 hours of untrimmed surveillance videos of expected events and 13 types of anomalies: *Abuse, Arrest, Arson, Assault, Accident, Burglary, Explosion, Fighting, Robbery, Shooting, Stealing, Shoplifting,* and *Vandalism*. The UCF-Crime dataset has originally been annotated at video level, but in UCFCrime2local it has been annotated at frame in *Vatic* format level by [21] and the latter has been used in our approach. The training dataset consists of 210 videos and the test set of 90 videos with only 7 classes out of the original classes of the UCF-Crime dataset: *Arrest, Assault, Burglary, Robbery, Stealing, Vandalism* and *Normal*. For normal videos in the dataset, all frames were considered normal. While frames with *lost flags* equal to 0 were considered as *hard anomalies* and others were considered *soft anomalies*, since the anomaly is not clear. Figure 5 shows the distribution of video in each of the classes considered. As we can see the unbalancing distribution where the normal videos is the majority class, while the other classes they are among there balanced with a difference at most of 3.5%. Such factor is relevant and must be taken in consideration during the training process to avoid biased model.

4.1.2 Autoencoder dataset

The dataset used to train the autoencoder for the privacy-preserving part is extracted from the UCF-Crime dataset. Specifically, for each training class, *Arrest, Assault, Burglary, Robbery, Stealing, Vandalism* and *Normal* are selected, in a random way, one video to have a good representation of each event. Figure 6 shows the distribution of the frames used to train the autoencoder. We can notice that we have a balancing distribution of the classes with a unique unbalancing towards the normal event. In such a case, the dataset unbalancing does not impact the model bias since the training is not focused on the class prediction but on the frame reconstruction.

4.2 Privacy Preserving using autoencoders and Differential Privacy

Autoencoder [44] are used to preprocess the video input frames to the end to remove sensitive information from the scene and build a mechanism which guarantees a training without model memorization. To reach this objective we experimented three different Differential Private (DP) [48] training optimizers with autoencoder neural networks and compared with the standard optimizer (Adam chosen as reference optimizer without differential privacy). Training with differential privacy provide strong privacy guarantees that the used algorithms learns only general patterns, and help to mitigate sensitive data expose risks. DP optimizers modify the gradients used by the optimization algorithm in two steps: (1) bounding the influence each input frame has on the gradients computation and adding random noise to these clipped gradients during gradient computation, so that it becomes statistically impossible to know if a particular data frame was included in the training dataset or not [4]. The list of optimizers experimented is reported in the following. These optimizers were released in TensorFlow, an open-source data flow engine released by Google.

[4] https://www.tensorflow.org/responsible_ai/privacy/tutorials/classification_privacy

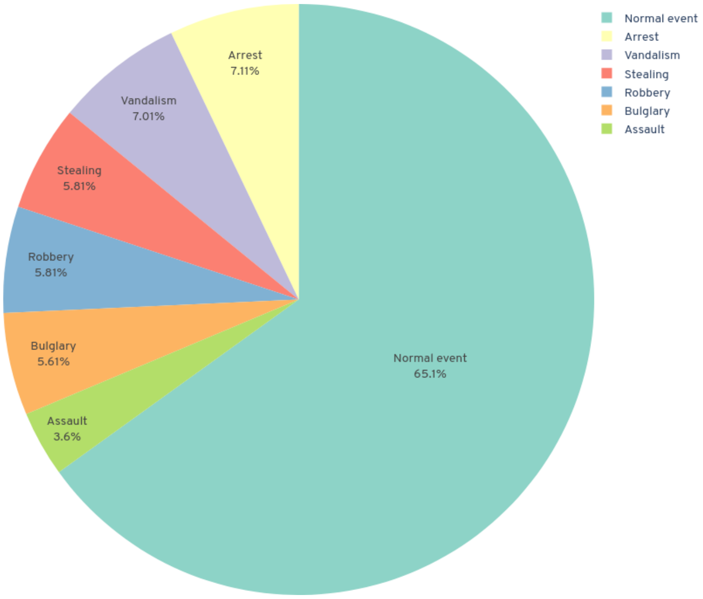


Figure 5: Video distribution Anomaly Detection

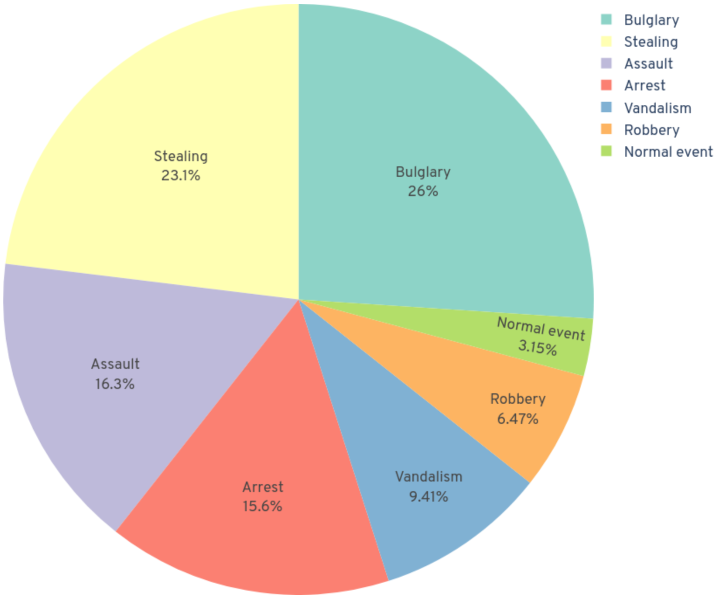


Figure 6: Frames distribution autoencoder

- Adaptive Moment Estimation (Adam) optimizer (AdamOpt) [5]: this optimizer computes exponential average weights of previous gradients and exponential average weights of previous gradients squares, applies bias correction mechanism on these weights, and these weights are updated at each iteration.

- Differential Private Stochastic Gradient Descent (SGD) optimizer (DPSGDOpt)[6]: is a differential private replacement of SGD optimizer. In the original SGD optimizer, a sample data frame denoted

[5] https://www.tensorflow.org/api_docs/python/tf/keras/optimizers/Adam

[6] https://www.tensorflow.org/responsible_ai/privacy/api_docs/python/tf_privacy/DPKerasSGDOptimizer

is randomly selected from the dataset to compute gradients of the loss function at each iteration of the training for the whole dataset. DPSGDOpt modifies the working schema of SGD by performing gradients computation for sample data frames with gradient normalization and averaging, then a random noise is added for privacy protection.

- Differential Private Adam optimizer (DPAdamOpt)[7]: this optimizer replaces the origional version of Adam optimizer using standard Gaussian noise insertion and by performing gradients computation for sample data frames with gradient normalization and averaging.
- Differential Private Adagrad optimizer (DPAdagOpt)[8]:this optimizer replaces the origional version of Adagrad optimizer. Adagrad optimizer work by giving slow learning rates for gradients with frequent updates and fast learning rates for gradients with infrequent updates, then it sums the squares of previous gradients and divides it by a specific high or low value. DPAdagOpt modifies this schema using standard Gaussian noise insertion and by performing gradients computation for sample data frames with gradient normalization and averaging.

For each DP optimizer we have varied the noise perturbation, from 0.1 (minumum perturbation) to 1.0 (maximum perturbation). It correspond to the ratio of the standard deviation to the clipping gradient. It is used to control how much noise is sampled and added to gradients before they are applied by the optimizer. Generally, more noise results in better privacy.

Training details The training has been performed, splitting the dataset randomly into a training set (70% of data), validation set (15%), and testing set (15%). The model has been trained, providing blocks of 50 frames and arresting the learning as soon as the validation loss increases or remains stable for five epochs. At the end of each processed block, the SSIM and PSNR have been evaluated. The overall training has been stopped whenever the SSIM and PSNR remained stable or decreased for six consecutive blocks. The learning rate has been set to 10^{-5}.

4.3 Advanced Techniques for Anomaly Detection Improvement

In the following section every advanced techniques applied for increase the efficiency and effectiveness of data utilization during the anomaly detection training are presented.

4.3.1 Impact of Bag-Of-Object on Anomaly Detection

Contextual information represented by the bag-of-objects and extracted by the object detector positively affect the detection of anomalies. We used four types of models in the experiments as below:

1. 2D-two-stream-CNN model with bag-of-objects used (***2S-bag***).
2. 2D-two-stream-CNN model without bag-of-objects (***2S-no-bag***).
3. Single flow CNN model for reconstructed frames only and with bag-of-objects used (***1S-bag***).
4. Single flow CNN without bag-of-objects (***1S-no-bag***).

Training Details: Prior training phase, several preprocessing steps are carried out: (1) dataset image resize into 224×224 and conversion from RGB to BGR, (2)zero-centering all color channels with respect to the ImageNet dataset and without the use of scaling, (3) test set videos are divided by half to create a validation set while preserving class distribution. For experimentation, the following rules are applied:

[7] https://www.tensorflow.org/responsible_ai/privacy/api_docs/python/tf_privacy/DPKerasAdamOptimizer

[8] https://www.tensorflow.org/responsible_ai/privacy/api_docs/python/tf_privacy/DPKerasAdagradOptimizer

1. An early stopping technique is used with the validation set to control model overfitting.
2. For video segments containing anomalies, the sampling frame rate was set to the rate of 3 FPS and for normal segments it was set to 10 FPS, normal segments in this context means segments that do not contain hard anomalies.
3. Hard sample mining has been used with $\alpha = 3$.
4. class weights were set to 1 for normal class and 2 for anomaly class.
5. Optimizers used are Adam, DPAdam, and DPAdag.
6. For each model were initially trained only the final fully-connected layers using a learning rate of 10^{-5} until reaching the lowest loss on validation set with the patience of 5 epochs. Then the networks were fully trained using the same methodology but with a learning rate of 10^{-6}. After each fully connected layer was applied a dropout of 0.5.

4.3.2 Impact of Hard Sample Mining on Anomaly Detection

This experiment investigates the impact of the Hard Sample Mining data balancing strategy on the anomaly detection classification problem. The models used for comparison are as below:

1. 2D-two-stream-CNN with hard sample mining strategy and with bag-of-objects used (***2S-bag***).
2. 2D-two-stream-CNN version trained without the use of balancing strategy and (***2S-bag-no-hm***)

For this experiment, all other training parameters were left unchanged with a low learning rate and early stopping to ensure fairness and an optimal degree of accuracy. Also, considering an epoch contains $\frac{1}{\alpha}$ of the samples of a training epoch that do not exploit it with hard sample mining strategy.

4.3.3 Impact of Soft Anomalies on Anomaly Detection

This experiment investigates the effect of soft anomalies explained in section XXX on model's performance during training phase. Using the same methodology of the experiment in previous section, two models were used in this experiment: *2S-bag* and *1S-bag* models trained without taking into consideration reconstructed frames with hard anomalies nor reconstructed normal frames. These frames are excluded frame the training process based on frames annotations, meaning that the frames with the an annotation *lost flag* equals 1 are excluded. But, no frames are being excluded from the test set and frames with soft anomalies are considered normal in order to ensure fair comparison with the other experiments. We will refer to the models trained with only hard anomalies as ***2S-bag-ha*** and ***1S-bag-ha***.

4.4 Privacy on Anomaly Detection

The anomaly detection on the video streaming is performed by a remote server which analyzes the frames passed by the smart camera as presented in section 3. The smart camera encodes the frame and passes to the server the latent space. The server decodes the frame and analyzes the reconstructed image. Before passing the frames to the server, compression and noise are applied. They provide a reconstruction frame with a certain level of noise needed to reduce the understanding level of sensitive information, e.g., faces. To demonstrate the privacy introduced by our method have been computed the frame similarity between the original image and the reconstructed one with the metrics presented in section 5. In addition, we experimented the contextual information detected into a reconstructed frame. Specifically, we applied a face detection algorithm, DeepFace [64], on the original and reconstructed frame, and we compared them. The frame quality reduction helps us to increase the privacy hiding the sensitive information, but it directly impacts anomaly detection accuracy. Therefore we conducted different experiments to find the best trade-off between privacy and accuracy. For example, increasing the noise in the DP training optimizer increases privacy at the expense of accuracy.

5 Experimental Results

In this section we discuss the results of the experiments previously described in section 4.

5.1 Evaluation Metrics

To measure the validity of our approach, we used two sets of metrics that are used (i) to demonstrate the privacy-preserving of our framework and (ii) to evaluate the anomaly detection.

5.1.1 Privacy preserving Evaluation Metrics

The evaluation of the privacy-preserving technique is performed by evaluating the quality degree of the frames after the application of the autoencoder. The main metrics used to evaluate the quality of an image [65] and used in our experiments are the following:

- **MSE (Mean Squared Error)** it measures the average of the squares of the errors, that is, the average squared difference between the estimated values and the actual value. This is the metric that provides the error calculation of the training model. However, it does not give no information about the quality of the image. For example, if every pixel were ”1” off, the picture would look brighter. However, a frame with half of the pixels ”1” off would look noisy but have a smaller MSE. The MSE is calculated as follow:

  $$MSE = \frac{\sum_{i=1}^{n}(x_i - \bar{x_i})^2}{n}$$

  where x_i is the predicted pixel, $\bar{x_i}$ is the actual value and n is the total amount of pixels.

- **PSNR (Peak Signal to Noise Ratio)** The signal-to-noise ratio measures the actual proportion of noise to the maximum possible value of a pixel. This is on a logarithmic scale, so minor differences will not factor in as much, and the final number mostly is how far off-peak differences are. Low values of PSNR correspond to a complete noise image, while increasing the PSNR value increases the quality of the image. The PSNR is calculated as follow:

  $$PSNR = 20 \times \log_{10}(\frac{max(I)}{\sqrt{MSE}})$$

  where $max(I)$ is the maximum value of the pixel in the image.

- **SSIM (Structural Similarity Index Measure)** This measure takes into account differences in luminance, contrast, and structure. The lower the SSIM value, the poorer the image quality.

5.1.2 Anomaly Detection Evaluation Metrics

As in [5] and [21] the various models were compared using the ***area under the ROC curve*** calculated frame by frame (FL-AUC). In the case of the 3D-CNN model, the classification is considered on the last frame of the segment as the model is intended to be used online. It was also introduced a new metric to evaluate the effectiveness of the models in the classification at the video-level: the ***video-level F_1 score*** (VL-F_1). This metric is useful to evaluate the models' anomaly detection capabilities with a weakly-labeled test set. The same model can obtain different results for the two metrics. For example, it can distinguish normal videos from anomalous ones correctly but not normal and anomalous segments inside the same video. It is therefore important to use both metrics to have a correct comparison. Using the VL-F_1 score, a whole video is classified as anomalous if its *sigmoidal anomaly score* exceeds 0.5 at any point and normal otherwise. Once all videos have been classified, the F_1 score is calculated as:

$$VL\text{-}F_1 = \frac{2}{\frac{1}{vl\text{-}r} + \frac{1}{vl\text{-}p}}$$

Where *vl-r* is the *video-level recall* and *vl-p* the *video-level precision*. SAS parameters have been heuristically set with: $\tau = 0.5$, $\nu = 2$, $UB = 7$ and $LB = -7$.

5.2 Evaluation of privacy preserving using autoencoder

In this section, we reported the results related to the privacy-preserving mechanism proposed using the autoencoder described in section 3. Specifically, we have shown the results obtained in the reconstruction frames, and we proved the effectiveness of autoencoder as a privacy-preserving mechanism showing the face detection applied on the reconstruction frames.

5.2.1 Autoencoder with DP

Figure 7 shows a comparison between the original frame and its decoder reconstruction using different training mechanisms. We selected a frame composed explicitly of sensitive information (person in front of the camera). As described in section 4.2, we experimented 4 different autoencoder training using different optimizers. In the absence of DP, we obtained a reconstruction frame visually near the original one. It is highlighted by SSIM, PSNR, and MSE, with values respectively 0.79 and 70.58 and 0.02. The worst reconstruction is provided by the DP Adam and the DP SGD optimizers. They are trained to add a noise factor of 0.1, and the result is relatively similar visually and statistically. They provide respectively SSIM 0.44 and 0.47, PSNR 65.44 and 65.25, and MSE 0.06. The best result is reached with the DP Adagrad optimizer, which provides an SSIM equal to 0.6, PSNR 66.85, and MSE equal to 0.04. As a result, we recognize that there are several people in front of the camera, but we cannot distinguish their identities. The MSE, as explained in section 5.1, it gives no information about the quality of the image. In fact, the value of MSE is low in each comparison.

5.2.2 Privacy Preserving in Reconstructed Frames

As explained in section 4, to prove the effectiveness of the privacy-preserving mechanism, we compared the face detection capacity of *DeepFace* on the original frame and on the reconstructed frames. Figure 8 shows the results obtained after the application of the face detection on the original frame. The model is able to detect 8 faces over 13. The undetected faces are due to the occlusions and brightness of the image. Instead, we notice that the faces detected considerably decrease when applying the detector on the frame reconstructed by the autoencoder trained with the Adam optimizer. Specifically, as shown on figure 9, 1 single face is detected over 13. Performing the same experiment on the other autoencoders

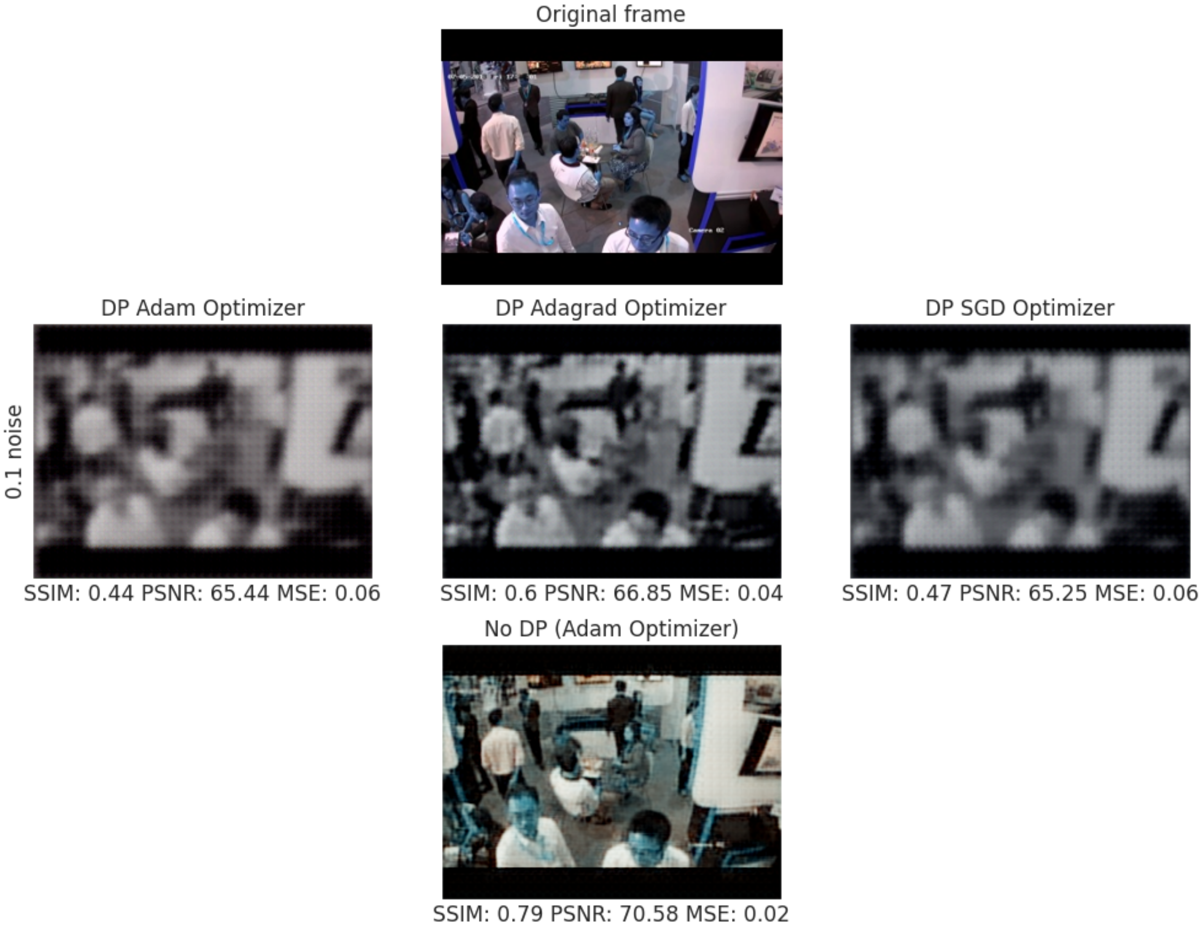

Original frame
DP Adam Optimizer
DP Adagrad Optimizer
DP SGD Optimizer
0.1 noise
SSIM: 0.44 PSNR: 65.44 MSE: 0.06
SSIM: 0.6 PSNR: 66.85 MSE: 0.04
SSIM: 0.47 PSNR: 65.25 MSE: 0.06
No DP (Adam Optimizer)
SSIM: 0.79 PSNR: 70.58 MSE: 0.02

Figure 7: Reconstruction frames using different training optimizers

trained with DP Adam, DP Adagrad, DP SGD optimizers, we obtained 0 faces detected over 13. Such results prove how the application of DP and noise addition contribute to making the sensitive information of the reconstruction frame, i.e., faces, not only visually unrecognizable, but also unrecognizable by a machine learning mechanism.

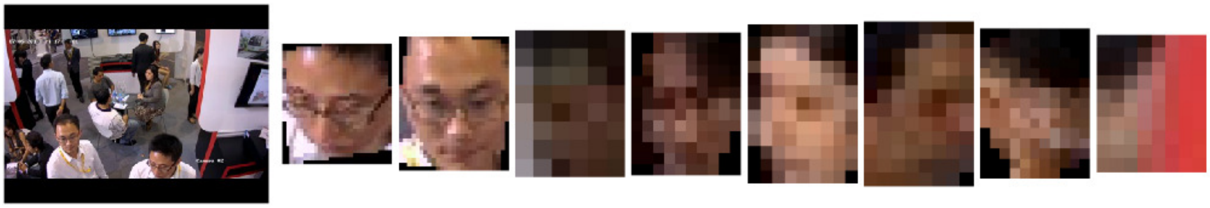


Figure 8: Faces detected in the original image

5.3 Impact of the advanced techniques on anomaly detection

In this section are exposed the anomaly detection results obtained applying the advanced training techniques described in section 4. **Bag-of-Objects.** Referring to Table 1 is possible to notice how the use of bag-of-objects has a clear impact on the model's ability to classify at frame-level. In particular, both

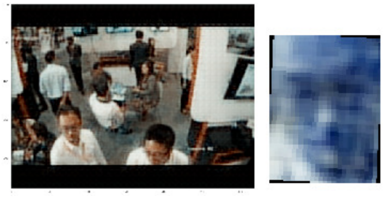


Figure 9: Faces detected in the reconstructed frame (Adam Optimizer)

Model	FL-AUC (%)	VL-F_1 (%)
2S-bag	82.6	66.7
2S-no-bag	77.6	61.5
2S-bag-no-hm	80.2	64.0
2S-bag-ha	**83.7**	**91.0**
1S-bag	78.6	74.1
1S-no-bag	75.3	58.3
1S-bag-ha	**81.5**	**87.5**

Table 1: Experimental Results

two-stream and one-stream architectures have an AUC increase of 6.4% and 4.4%: from 77.6% (2S-no-bag) to 82.6% (2S-bag) and from 75.3% (1S-no-bag) to 78.6% (1S-bag). Also, with regard to the video-level F_1 score both architectures have an increase of 8.5% and a remarkable 27.1%: from 61.5% (2S-no-bag) to 66.7% (2S-bag) and from 58.3% (1S-no-bag) to 74.1% (1S-bag). It certifies that the use of the bag-of-object has a positive impact on the classification of the single frames and the distinction of videos containing anomalies from normal ones.

Hard Mining. Always referring to Table 1 the impact of hard mining on anomalous detection can be verified by comparing 2S-bag with 2S-bag-no-hm. There is an increase in AUC accuracy of 3%, from 80.2% to 82.6%, with the use of hard mining and also the VL-F_1 score increases by 4.2%, from 64.0% to 66.7%.

Soft Anomalies. In Table 1 can also be assessed the impact of the exclusive use of *hard anomalies* as positive samples. For both architectures the maximum AUC values are reached, with an increase of 1.3% and 3.7%: from 82.6% (2S-bag) to 83.7% (2S-bag-ha) and from 78.6% (1S-bag) to 81.5% (1S-bag-ha). But the most significant results are the VL-F_1 values with a significant increase of 36.4% and 18.1%: from 66.7% (2S-bag) to 91.0% (2S-bag-ha) and from 74.1% (1S-bag) to 87.5% (1S-bag-ha). It means that an accurate selection of abnormal and normal frames – so that they are completely unambiguous for their use in training – leads to a general improvement of the frame-level classification and allows a considerably better video-level detection.

5.4 Impact of privacy preserving techniques on anomaly detection

The impact of privacy-preserving (autoencoder reconstruction frame) on the anomaly detection is compared using the Area Under the Curve (ROC). Figure 10 shows the comparison between three anomaly detection models trained with the frames encoded by the client-side and decoded on the server-side. As autoencoder model, we considered the model without DP and Adam optimizer (*No DP*), and two models with DP. Specifically, the model trained with DP Adam optimizer (*DP Adam Opt*), where we obtained the worst reconstruction frame, and the model trained with DP Adagrad optimizer (*DP Adagrad Opt*), through which we obtained the best reconstruction frame with DP. The model with the *DP SGD* optimizer is not taken into consideration since it produces an image reconstruction quality similar to the DP Adam optimizer. As the ROC curve shows, the reconstruction frames' quality has a high correlation with the anomaly detection accuracy. The best reconstruction quality frame, obtained without DP, produces the highest AUC (0.788). Decreasing the quality of the frame (DP Adagrad optimizer), we obtained a decrement of detection accuracy (AUC = 0.707). Finally, we obtained the worst detection accuracy with the worst frame quality, represented by AUC equal to 0.577. To summarize, we can claim that the privacy-preserving increment produces an anomaly detection reduction because the quality of the reconstruction frame is not sufficient to distinguish contextual information from the scene. Hence a trade-off between anomaly detection accuracy and privacy-preserving is indispensable.

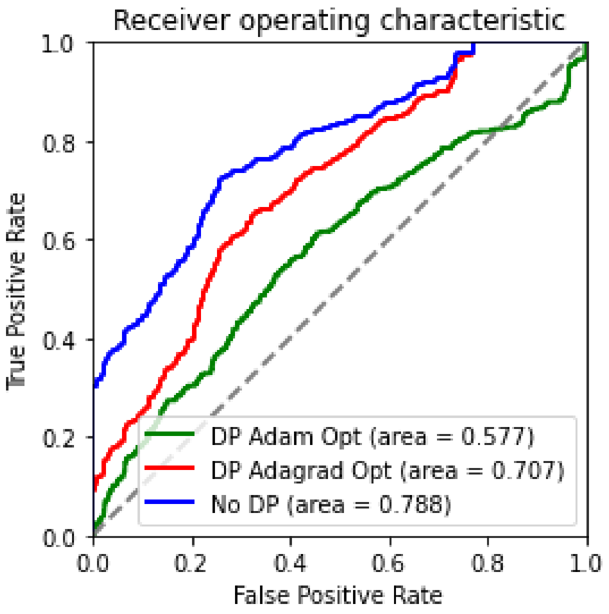


Figure 10: Anomaly Detection with Autoencoder frame reconstruction

6 Conclusion and Future Work

New challenges are arising in the latest years concerning multimedia data analysis. The high availability of data and the possibility to perform analysis online, thanks to the increased connectivity speed and the possibility to perform computationally challenging operations also on constrained devices, make it possible to provide new services, to analyze and timely handle emergency situations. However, being able to perform this analysis in a privacy-preserving way, i.e., without violating the privacy of the people involved in pictures and videos, would make it possible to apply these technologies virtually in any environment, without violating international privacy regulations. In this paper, we proposed a novel approach to perform privacy-preserving anomaly detection in video-frames, exploiting deep learning analysis techniques to identify anomalous and potentially dangerous situations involving violence or other activities that can cause harm to people. By the usage of autoencoders with differential privacy, we have ensured that the performed analysis is done in a privacy-preserving way. We have experimentally verified that the application of autoencoders to anonymize and reconstruct video frames marginally affects the capability of identifying anomalies still it strongly affects the capabilities of recognising the identity of people present in each video frame.

Envisioned future directions for this work include training existing object detection and image classification algorithms on anonymized image datasets generated by autoencoders to increase model accuracy. In addition, the actual object recognition model is trained with very generic classes, which puts a limit for the model. To overcome such limitation, the contextual information given by the bag-of-object can be enriched considering also more relevant objects for the specific anomaly detection task (i.e., knives, guns). Another direction would be the usage and evaluation of explainability indexes to enable the verification of the parameters used to take decisions and their compatibility with ethical standards. Further future directions would be the inclusion of features related to the objects present in the video and their relation with the monitored people. Finally, the application on live captured video for real-time detection is considered a possible further extension.

Acknowledgments

This work has been partially funded by EU funded project H2020 SIFIS-Home GA ID:952652.

References

[1] E. Ahmed, M. Jones, and T.K. Marks. An improved deep learning architecture for person re-identification. In *Proc. of the 23rd IEEE Conference on Computer Vision and Pattern Recognition (CVPR'15), Boston, Massachusetts, USA*, pages 3908–3916. IEEE, June 2015.

[2] G. Giorgi, A. Saracino, and F. Martinelli. Using recurrent neural networks for continuous authentication through gait analysis. *Pattern Recognition Letters*, 147:157–163, July 2021.

[3] T. Fernando, S. Denman, S. Sridharan, and C. Fookes. Tracking by prediction: A deep generative model for mutli-person localisation and tracking. In *Proc. of the 18th IEEE Winter Conference on Applications of Computer Vision (WACV'18), Lake Tahoe, Nevada, USA*, pages 1122–1132. IEEE, March 2018.

[4] A. Akula, A.K. Shah, and R. Ghosh. Deep learning approach for human action recognition in infrared images. *Cognitive Systems Research*, 50:146–154, August 2018.

[5] W. Sultani, C. Chen, and M. Shah. Real-world anomaly detection in surveillance videos. In *Proc. of the 26th IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'18), Salt Lake City, Utah, USA*, pages 6479–6488. IEEE, June 2018.

[6] Z. Ren, Y. Jae Lee, and M.S. Ryoo. Learning to anonymize faces for privacy preserving action detection. In *Proc. of the 15th european conference on computer vision (ECCV'18), Munich, Germany*, volume 11205 of *Lecture Notes in Computer Science*, pages 620–636. Springer, Cham, September 2018.

[7] S. Petrocchi, G. Giorgi, and M.G.C.A. Cimino. A real-time deep learning approach for real-world video anomaly detection. In *Proc. of the 16th International Conference on Availability, Reliability and Security (ARES'21), Vienna, Austria*, pages 1–9. ACM, August 2021.

[8] F. Jiang, J. Yuan, S.A. Tsaftaris, and A.K. Katsaggelos. Anomalous video event detection using spatiotemporal context. *Computer Vision and Image Understanding*, 115(3):323–333, March 2011.

[9] C. Li, Z. Han, Q. Ye, and J. Jiao. Visual abnormal behavior detection based on trajectory sparse reconstruction analysis. *Neurocomputing*, 119:94–100, November 2013.

[10] R.V.H.M. Colque, C. Caetano, M.T.L.D. Andrade, and W.R. Schwartz. Histograms of optical flow orientation and magnitude and entropy to detect anomalous events in videos. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(3):673–682, March 2017.

[11] M.J. Roshtkhari and M.D. Levine. Online dominant and anomalous behavior detection in videos. In *Proc. of the 21st IEEE conference on computer vision and pattern recognition (CVPR'13), Portland, Oregon, USA*, pages 2611–2618. IEEE, October 2013.

[12] R. Mehran, A. Oyama, and M. Shah. Abnormal crowd behavior detection using social force model. In *Proc. of the 17th IEEE Conference on Computer Vision and Pattern Recognition (CVPR'09), Miami, Florida, USA*, pages 935–942. IEEE, June 2009.

[13] W. Li, V. Mahadevan, and N. Vasconcelos. Anomaly detection and localization in crowded scenes. *IEEE transactions on pattern analysis and machine intelligence*, 36(1):18–32, January 2013.

[14] A. Krizhevsky, I. Sutskever, and G.E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proc. of the 25th International Conference on Neural Information Processing Systems (NIPS'12), Lake Tahoe, Nevada, USA*, pages 1097–1105. ACM, December 2012.

[15] M. Hasan, J. Choi, J. Neumann, A.K. Roy-Chowdhury, and L.S. Davis. Learning temporal regularity in video sequences. In *Proc. of the 24th IEEE conference on computer vision and pattern recognition (CVPR'16), Las Vegas, Nevada, USA*, pages 733–742. IEEE, June 2016.

[16] M. Mirza and S. Osindero. Conditional generative adversarial nets. arXiv:1411.1784, 2014. `https://doi.org/10.48550/arXiv.1411.1784`.

[17] W. Liu, W. Luo, D. Lian, and S. Gao. Future frame prediction for anomaly detection–a new baseline. In *Proc. of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR'18), Salt Lake City, Utah, USA*, pages 6536–6545. IEEE, June 2018.

[18] D. Gong, L. Liu, V. Le, B. Saha, M. R. Mansour, S. Venkatesh, and A.V.D. Hengel. Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In *Proc. of the 17th IEEE/CVF International Conference on Computer Vision (ICCV'19), Seoul, South Korea*, pages 1705–1714. IEEE, November 2019.

[19] S. Bhakat and G. Ramakrishnan. Anomaly detection in surveillance videos. In *Proc. of the 6th ACM India Joint International Conference on Data Science and Management of Data (CoDS-COMAD'19), Kolkata, India*, pages 252–255. ACM, January 2019.

[20] R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proc. of the 16th IEEE International Conference on Computer Vision (ICCV'17), Venice, Italy*, pages 618–626. IEEE, October 2017.

[21] F. Landi, C.G.M. Snoek, and R. Cucchiara. Anomaly locality in video surveillance. arXiv:1901.10364, January 2019. `https://doi.org/10.48550/arXiv.1901.10364`.

[22] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proc. of the 2015 IEEE International conference on Computer Vision (ICCV'15), Santiago, Chile*, pages 4489–4497. IEEE, December 2015.

[23] Y. Zhu, X. Li, C. Liu, M. Zolfaghari, Y. Xiong, C. Wu, Z. Zhang, J. Tighe, R. Manmatha, and M. Li. A comprehensive study of deep video action recognition. arXiv:2012.06567, December 2020. `https://doi.org/10.48550/arXiv.2012.06567`.

[24] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Proc. of the 25th IEEE Conference on Computer Vision and Pattern Recognition (CVPR'17), Honolulu, Hawaii, USA*, pages 6299–6308. IEEE, July 2017.

[25] J. Henrio and T. Nakashima. Anomaly detection in videos recorded by drones in a surveillance context. In *Proc. of the 31st IEEE International Conference on Systems, Man, and Cybernetics (SMC'18), Miyazaki, Japan*, pages 2503–2508. IEEE, October 2018.

[26] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. arXiv:1406.2199, November 2014. `https://doi.org/10.48550/arXiv.1406.2199`.

[27] M.S. Ryoo, B. Rothrock, C. Fleming, and H.J. Yang. Privacy-preserving human activity recognition from extreme low resolution. In *Proc. of the 31st AAAI Conference on Artificial Intelligence (AAAI'17), San Francisco, California, USA*, pages 4255–4262. AAAI Press, February 2017.

[28] P. Aditya, R. Sen, P. Druschel, S.J. Oh, R. Benenson, M. Fritz, B. Schiele, B. Bhattacharjee, and T.T. Wu. I-pic: A platform for privacy-compliant image capture. In *Proc. of the 14th annual international conference on mobile systems, applications, and services (MobiSys'16), Singapore, Singapore*, pages 235–248. ACM, June 2016.

[29] J. Wang, B. Amos, A. Das, P. Pillai, N. Sadeh, and M. Satyanarayanan. A scalable and privacy-aware iot service for live video analytics. In *Proc. of the 8th ACM on Multimedia Systems Conference (MMSys'17),*

Taipei, Taiwan, pages 38–49. ACM, June 2017.

[30] D.J. Butler, J. Huang, F. Roesner, and M. Cakmak. The privacy-utility tradeoff for remotely teleoperated robots. In *Proc. of the 10th ACM/IEEE international conference on human-robot interaction (HRI'15), Portland, OR, USA*, pages 27–34. IEEE, March 2015.

[31] C. Neustaedter and S. Greenberg. The design of a context-aware home media space for balancing privacy and awareness. In *Proc. of the 5th International Conference on Ubiquitous Computing (UBICOMP'03), Seattle, Washington, USA*, volume 2782 of *Lecture Notes in Computer Science*, pages 297–314. Springer-Verlag, October 2003.

[32] C. Zhang, Y. Rui, and L. He. Light weight background blurring for video conferencing applications. In *Proc. of the 13th International Conference on Image Processing (ICIP'06), Atlanta, Georgia, USA*, pages 481–484. IEEE, October 2006.

[33] M. Boyle, C. Edwards, and S. Greenberg. The effects of filtered video on awareness and privacy. In *Proc. of the 3rd ACM conference on Computer supported cooperative work (CSCW'00), Philadelphia, Pennsylvania, USA*, pages 1–10. ACM, December 2000.

[34] I. Kitahara, K. Kogure, and N. Hagita. Stealth vision for protecting privacy. In *Proc. of the 17th International Conference on Pattern Recognition (ICPR'04), Cambridge, United Kingdom*, pages 404–407. IEEE, August 2004.

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

[36] T. Winkler, A. Erdélyi, and B. Rinner. Trusteye. m4: protecting the sensor—not the camera. In *Proc. of the 11th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS'14), Seoul, South Korea*, pages 159–164. IEEE, August 2014.

[37] Y. Zhong, R. Arandjelović, and A. Zisserman. Faces in places: Compound query retrieval. In *Proc. of the 27th British Machine Vision Conference (BMVC'16), York, United Kingdom*, pages 1–12. BMVA Press, September 2016.

[38] E.M. Newton, L. Sweeney, and B. Malin. Preserving privacy by de-identifying face images. *IEEE transactions on Knowledge and Data Engineering*, 17(2):232–243, February 2005.

[39] R Gross, E. Airoldi, B. Malin, and L. Sweeney. Integrating utility into face de-identification. In *Proc. of the 5th International Workshop on Privacy Enhancing Technologies (PET'05), Cavtat, Croatia*, volume 3856 of *Lecture Notes in Computer Science*, pages 227–242. Springer, Berlin, Heidelberg, May 2005.

[40] M. Granados, J. Tompkin, K. Kim, O. Grau, J. Kautz, and C. Theobalt. How not to be seen—object removal from videos of crowded scenes. *Computer Graphics Forum*, 31(2):219–228, May 2012.

[41] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. In *Proc. of the 27th annual conference on Computer graphics and interactive techniques (SIGGRAPH'00), New Orleans, Louisiana, USA*, pages 417–424. ACM, July 2000.

[42] A.C. Kokaram, R.D. Morris, W.J. Fitzgerald, and P.J.W. Rayner. Interpolation of missing data in image sequences. *IEEE Transactions on Image Processing*, 4(11):1509–1519, November 1995.

[43] A.R. Abraham, A.K. Prabhavathy, and J.D. Shree. A survey on video inpainting. *International Journal of Computer Applications*, 56(9):43–47, June 2012.

[44] D. Bank, N. Koenigstein, and R. Giryes. Autoencoders. arXiv:2003.05991, April 2020. `https://doi.org/10.48550/arXiv.2003.05991`.

[45] M. D'Souza et al. Autoencoder-a new method for keeping data privacy when analyzing videos of patients with motor dysfunction (p4. 001). *Neurology*, 90(15 Supplement), 2018.

[46] M. Malekzadeh, R.G. Clegg, and H. Haddadi. Replacement autoencoder: A privacy-preserving algorithm for sensory data analysis. In *Proc. of the 3rd IEEE/ACM International Conference on Internet-of-Things Design and Implementation (IoTDI'18), Orlando, Florida, USA*, pages 165–176. IEEE, April 2018.

[47] O. Hajihassani, O. Ardakanian, and H. Khazaei. Latent representation learning and manipulation for privacy-preserving sensor data analytics. In *Proc. of the 2nd IEEE Workshop on Machine Learning on Edge in Sensor Systems (SenSys-ML'20), Sydney, Australia*, pages 7–12. IEEE, April 2020.

[48] C. Dwork. Differential privacy: A survey of results. In *Proc. of the 5th International conference on theory and applications of models of computation (TAMC'08), Xi'an, China*, volume 4978 of *Lecture Notes in Computer*

Science, pages 1–19. Springer, Berlin, Heidelberg, April 2008.

[49] L. Xie, K. Lin, S. Wang, F. Wang, and J. Zhou. Differentially private generative adversarial network. arXiv:1802.06739, February 2018. `https://doi.org/10.48550/arXiv.1802.06739`.

[50] X. Zhang, S. Ji, and T. Wang. Differentially private releasing via deep generative model (technical report). arXiv:1801.01594, March 2018. `https://doi.org/10.48550/arXiv.1801.01594`.

[51] M. Abadi, A. Chu, I. Goodfellow, H.B. McMahan, I. Mironov, K. Talwar, and L. Zhang. Deep learning with differential privacy. In *Proc. of the 23rd ACM SIGSAC conference on computer and communications security (CCS'16), Vienna, Austria*, pages 308–318. ACM, October 2016.

[52] P. Sirohi, A. Agarwal, and S. Tyagi. A comprehensive study on security attacks on ssl/tls protocol. In *Proc. of the 2nd International Conference on Next Generation Computing Technologies (NGCT'16), Dehradun, India*, pages 893–898. IEEE, October 2016.

[53] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. of the 24th IEEE conference on computer vision and pattern recognition (CVPR'16), Las Vegas, Nevada, USA*, pages 770–778. IEEE, December 2016.

[54] A. Bochkovskiy, C. Wang, and H.M. Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv*, 2004:1–17, April 2020.

[55] M. Malekzadeh, R.G. Clegg, A. Cavallaro, and H. Haddadi. Mobile sensor data anonymization. In *Proc. of the 2nd international conference on internet of things design and implementation (IoTDI'19), Montreal, Quebec, Canada*, pages 49–58. ACM, April 2019.

[56] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey. Adversarial autoencoders. arXiv preprint arXiv:1511.05644, May 2016. `https://doi.org/10.48550/arXiv.1511.05644`.

[57] J. Liu, J. Liu, P. Li, and Z. Kuang. Embedded autoencoders: A novel framework for face de-identification. In *Proc. of the 2nd International Cognitive Cities Conference (IC3'19), Kyoto, Japan*, pages 154–163. Springer, September 2019.

[58] A. Mosallanezhad, Y.N. Silva, M. Mancenido, and H. Liu. Toward privacy and utility preserving image representation. *arXiv preprint arXiv*, 2009:1–11, September 2020.

[59] M. Maire T. Lin, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C.L. Zitnick. Microsoft coco: Common objects in context. In *Proc. of the 13th European conference on computer vision (ECCV'14), Zurich, Switzerland*, volume 8693 of *Lecture Notes in Computer Science*, pages 740–755. Springer, Cham, September 2014.

[60] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proc. of the 17th IEEE conference on computer vision and pattern recognition (CVPR'09), Miami, Florida, USA*, pages 248–255. IEEE, June 2009.

[61] B.K.P. Horn and B.G. Schunck. Determining optical flow. *Artificial intelligence*, 17(1-3):185–203, August 1981.

[62] Q. Dong, S. Gong, and X. Zhu. Class rectification hard mining for imbalanced deep learning. In *Proc. of the 16th IEEE International Conference on Computer Vision (ICCV'17), Venice, Italy*, pages 1851–1860. IEEE, October 2017.

[63] K. Doshi and Y. Yilmaz. Online anomaly detection in surveillance videos with asymptotic bound on false alarm rate. *Pattern Recognition*, 114:107865–107873, June 2021.

[64] S.I. Serengil and A. Ozpinar. Lightface: A hybrid deep face recognition framework. In *Proc. of the 3rd Innovations in Intelligent Systems and Applications Conference (ASYU'20), Istanbul, Turkey*, pages 23–27. IEEE, October 2020.

[65] A. Hore and D. Ziou. Image quality metrics: Psnr vs. ssim. In *Proc. of the 20th international conference on pattern recognition (ICPR'10), Istanbul, Turkey*, pages 2366–2369. IEEE, August 2010.

Author Biography

Giacomo Giorgi received the B.S. and M.S. degrees in Computer Engineering from University of Pisa in 2012 and 2016, and Ph.D. degrees in the same University in 2021. Currently he is an research fellow in the Institute for Informatics and Telematics at National Council of Research in Pisa. His research interests include Behavioral analysis, Artificial Intelligence, Video analysis, Intrusion Detection System, Textual information analysis.

Wisam Abbasi received the B.S. degree in Computer Information Systems from An-Najah National University in 2011 and M.S. degree in Computing from Birzeit University in 2018. Currently she is a research fellow in the Institute for Informatics and Telematics at National Council of Research in Pisa and a PhD student at Computer Science Department of Pisa University. Her research interests include Artificial Intelligence, Privacy Preserving Data Analysis, Trustworthy AI, Interpretable (Explainable) AI.

Andrea Saracino (Ph.D. 2015, M.Eng. 2011) is a Researcher at Istituto di Informatica e Telematica of the National Research Council (IIT-CNR) of Italy. His research is focused on applications of AI for security of mobile and distributed systems, with an emphasis on intrusion and malware detection in Android devices. He is the co-chair of the working group on AI & Cybersecurity of the Italian Association for AI (AIxIA). He is the project coordinator for the H2020 project SIFIS-Home and he is (or has been) involved in a number of EU-project such as EU-H2020 E-Corridor, EU-H2020 C3ISP, EU-H2020 NeCS, EU-H2020 Cybersure, EIT-Digital Trusted Cloud and IoT.