frontiers in Signal Processing

ORIGINAL RESEARCH
published: 02 November 2021
doi: 10.3389/frsip.2021.765006

Deep Learning-Based Object Tracking *via* Compressed Domain Residual Frames

*Karim El Khoury**[†], *Jonathan Samelson*[†] *and Benoît Macq*

Institute of Information and Communication Technologies, Electronics and Applied Mathematics, Université catholique de Louvain, Louvain-la-Neuve, Belgium

The extensive rise of high-definition CCTV camera footage has stimulated both the data compression and the data analysis research fields. The increased awareness of citizens to the vulnerability of their private information, creates a third challenge for the video surveillance community that also has to encompass privacy protection. In this paper, we aim to tackle those needs by proposing a deep learning-based object tracking solution via compressed domain residual frames. The goal is to be able to provide a public and privacy-friendly image representation for data analysis. In this work, we explore a scenario where the tracking is achieved directly on a restricted part of the information extracted from the compressed domain. We utilize exclusively the residual frames already generated by the video compression codec to train and test our network. This very compact representation also acts as an information filter, which limits the amount of private information leakage in a video stream. We manage to show that using residual frames for deep learning-based object tracking can be just as effective as using classical decoded frames. More precisely, the use of residual frames is particularly beneficial in simple video surveillance scenarios with non-overlapping and continuous traffic.

Keywords: deep learning, video compression, residual frames, video surveillance, object detection, object tracking, privacy protection, HOTA

OPEN ACCESS

Edited by:
Lukas Esterle,
Aarhus University, Denmark

Reviewed by:
Stefania Colonnese,
Sapienza University of Rome, Italy
Ionut Schiopu,
Vrije University Brussel, Belgium

****Correspondence:***
Karim El Khoury
karim.elkhoury@uclouvain.be

[†]These authors have contributed equally to this work and share first authorship

Specialty section:
This article was submitted to Image Processing, a section of the journal Frontiers in Signal Processing

Received: *26 August 2021*
Accepted: *11 October 2021*
Published: *02 November 2021*

Citation:
El Khoury K, Samelson J and Macq B (2021) Deep Learning-Based Object Tracking via Compressed Domain Residual Frames. Front. Sig. Proc. 1:765006. doi: 10.3389/frsip.2021.765006

1 INTRODUCTION

According to Cisco's Visual Networking Index report in 2017, global data consumption has been increasing exponentially for the past decade, with video data accounting for 80% of the worldwide traffic[1]. One of the largest growing types of video data consumption is video surveillance traffic, which is set to achieve a seven-fold increase by 2022 to account for a total of 3% of the worldwide Internet traffic. This substantial surge in video surveillance data had created three major needs. The first major need is to be able to transfer and store the data which calls for the use of innovative video compression codecs. The second major need is to be able to analyze the large flow of data, which calls for the use of machine learning and more specifically deep learning algorithms. Lastly, the third major need, which is especially relevant when working with video surveillance footage, is to able to preserve the privacy of the individuals involved in the captured scenes. The main motivation of this paper is to address these three needs by using an inexpensive, low-storage and privacy-friendly image representation that can therefore be made publicly available for traffic analysis.

[1] https://twiki.cern.ch/twiki/pub/HEPIX/TechwatchNetwork/HtwNetworkDocuments/white-paper-c11-741490.pdf

In this work, we utilize an inexpensive compressed domain image representation already generated by the video compression codec: residual frames. Also known as the prediction error, residual frames are the difference between the prediction of a frame at time *t+1* using the frame at time *t* and the original frame at time *t+1*. Residual frames not only have a low-storage cost but they also act as an information filter by only keeping the movement regions of interest (ROI) between two consecutive frames. In this research, we choose to work exclusively on the residual frames to train and test a deep learning-based object detector and tracker. This will allow us to not only store our data in the compressed format but also provide a privacy-friendly data source for deep learning-based object tracking. Deep learning-based object tracking trained and tested solely on residual frames is a new approach explored by this paper. This research's main contribution is to show that using residual frames as an image representation for a deep learning-based object tracking can be just as effective as using decoded frames while limiting the amount of private information leakage in a video stream.

This paper is organized as follows. In **Section 2**, we detail the state of the art of combining data compression, data analysis and data protection. In **Section 3** we put forward the utilized materials and methods by presenting the compression algorithm and both object detectors and trackers used. Then we introduce the HOTA evaluation metric, the different datasets that we have worked with, and the two experiments that have been conducted. In **Section 4** we present the detailed individual results of the two experiments. Thereafter, in **Section 5**, we expose the benefits and drawbacks resulting from the two experiments, examine the privacy-friendly capabilities of our solution, and comment on key limitations that have impacted this research. Finally, in **Section 6**, we conclude the paper by summarizing the results and outcomes of our research and proposing several potential further work.

2 RELATED WORKS

The challenges of combining the two needs of video compression and video analysis is a topic that has already been addressed in the literature. The Moving Picture Experts Group (MPEG) recently created an ad hoc group dedicated to the standardization of Video Coding for Machines (VCM) (Duan et al., 2020). The VCM group's inception came after the realization that traditional video compression codecs were not optimal for deep learning feature extraction. The aim of the VCM group is to create a video compression codec tailored to machine vision rather than human perception. The proposed codec managed to achieve, at lower bit-rate costs, much better detection accuracy in most cases and more visually pleasing decoded videos than the High Efficiency Video Codec (HEVC). Another proposition within the same scope proposed a hybrid framework that combined convolutional neural networks (CNN) with classical background subtraction techniques (Kim et al., 2018). The proposed framework was made up of a two-step process. The first step was to identify the ROI using a background subtraction algorithm on all frames. The second step was to apply a CNN classifier to the ROI. They managed to achieve a classification accuracy of up to 85%.

In addition, other works have also looked at taking advantage of already generated compressed domain motion vectors to improve the efficiency of deep learning networks. Researchers proposed to work on a CNN-based detector combined with compressed domain motion vectors to lower the power consumption of classical deep learning-based detectors (Ujiie et al., 2018). They utilized the inexpensive motion vectors already generated by the video compression codec to speed up the detection process in the predicted frames. Using the MOT16 benchmark, they obtained a MOTA score of 88% while also cutting the detection frequency by twelve times. Other researchers have also explored the use of the compressed domain motion vectors, but concentrated their efforts on improving the efficiency of their CNN-based object tracker (Liu et al., 2019). They manage to achieve a tracker that is six times faster than the state-of-the-art online multi-object tracking (MOT) methods.

Another challenge that has been addressed in the literature is to combine the two needs of data analysis and data privacy. Researchers have tried to simplify the problem by proposing to tackle specific features and excluding them from the frames as a binary decision. An image scrambling method for privacy-friendly video surveillance showed that it was possible to scramble the frame's ROI to hide critical information in the observed scene (Dufaux and Ebrahimi, 2006). Other related work take up the challenge of combining data compression and data privacy. Researchers developed a custom license plate recognition and facial recognition software to encrypt the specific ROI before encoding and sending out the video sequence (Carrillo et al., 2008).

Although the presented works tackle at least one of the three major needs, none of them attempt to tackle all three major needs in one unified solution.

3 MATERIALS AND METHODS

In this section we present the compression algorithm used, the object detectors and trackers that we have worked with, our evaluation metric and datasets, and finally our experimental setup. All publicly available source codes used in this work are made available at https://github.com/JonathanSamelson/ResidualsTracking.

3.1 Compression Algorithm

Inter-frame video codecs use the temporal redundancies of a video sequence to compress it. This is achieved by segmenting the video sequences into reference frames (also called I frames) and predicted frames (also called P or B frames). The reference frames consist of sending the full intra-frame image whereas the predicted frames are generated by a process called block matching. Block matching divides the frame into several non-overlapping blocks of predetermined size and assigns a motion vector (also known as a displacement vector) to each block by

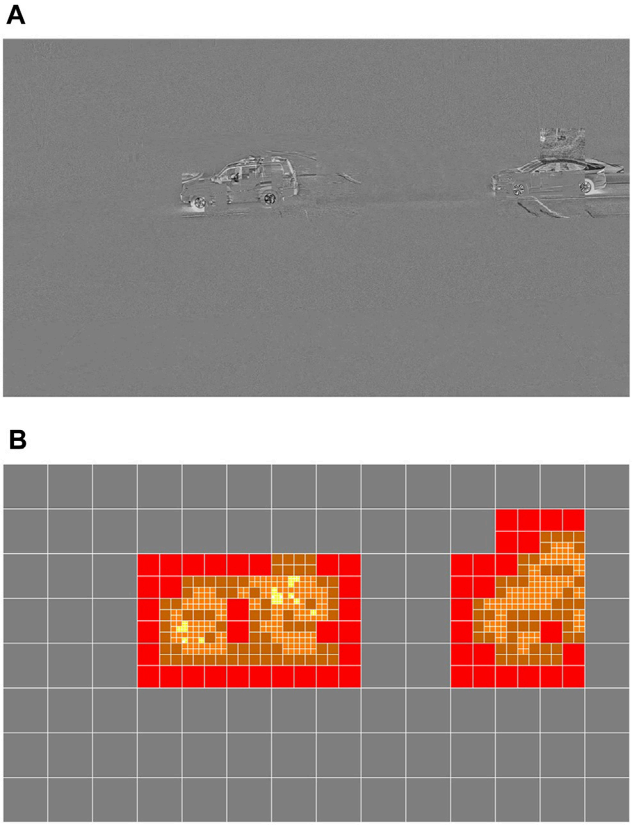

FIGURE 1 | Sample image of the residual frame **(A)** and the respective adaptive block matching quadtree **(B)** generated from two successive frames.

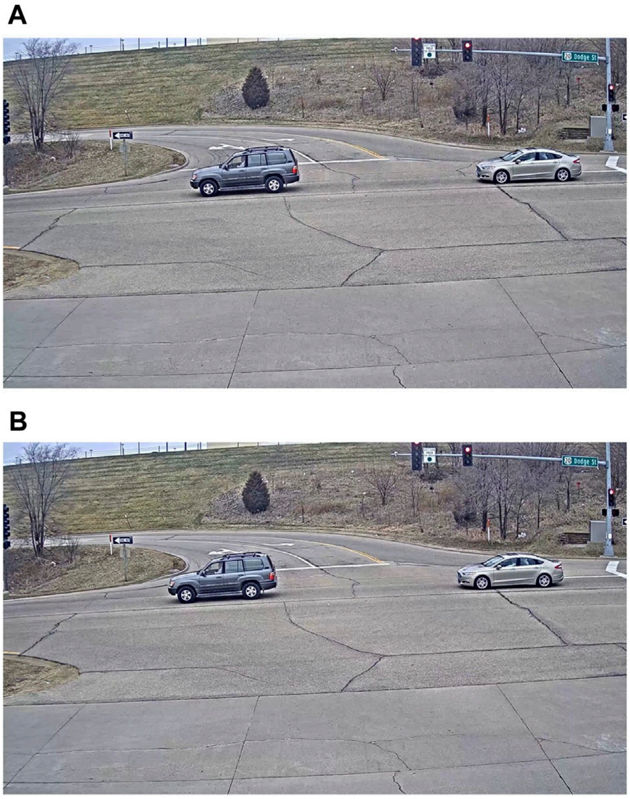

FIGURE 2 | Sample image of two successive decoded frames **(A)** and **(B)**.

identifying the location of that block in the previous frame. The motion vectors paired with the latest original stored frame enable us to make a prediction on the upcoming frame and subsequently generate the frame prediction error (also called residual frame) by subtracting the latest original frame from the predicted frame.

The most widely used inter-frame video compression formats such as HEVC (Sullivan et al., 2012), VVC (Huang et al., 2020), and VP9 (Mukherjee et al., 2013) all rely on a quadtree structure for their block matching process called adaptive block matching. The goal is to have variable block sizes that depend on the scene depicted in the frame. Ideally, we would like to have large blocks that represent inanimate sections of the frame for the background, and small blocks that represent movement areas, for the foreground. This would lower the encoding cost per frame, as it would reduce the number of encoded blocks and motion vectors per frame. The adaptive block matching process can also be seen as an image content filter given that it highlights ROI in the frame (movement areas) over the inanimate sections of the frame. **Figure 1** shows a sample image of the residual frame (A) as well as the respective adaptive block matching quadtree (B) generated for the two successive decoded frames shown in **Figure 2**.

For this work we developed our own open-source generic adaptive block matching algorithm inspired by the works of (Vermaut et al., 2001; Barjatya, 2004). Our algorithm works similarly to the widely used standardized inter-frame compression formats such as presented in (Chien et al., 2021; Zhang et al., 2019). This allows us to generate the motion vectors and corresponding residual frames needed for our study without accessing, editing, and testing all the available inter-frame compression video codecs. The algorithm needs to know only three preset values: the size of the largest possible block, the size of the smallest possible block, and the sensitivity threshold. The algorithm starts by calculating the motion vectors for the largest blocks. Once it has done so, it goes through the motion vectors individually from left to right and from top to bottom and looks at each block's individual neighbors. If the absolute value of the difference of the motion vector and the averages of its neighbors are greater than the preset threshold, the concerned block is split. Otherwise, this motion vector is confirmed and becomes final. The act of splitting means that the block containing the motion vector will be divided into four equal blocks and the process continues recursively until we reach the preset smallest block size. The detailed algorithm is shown in **Figure 3**.

3.2 Object Detectors

In this work we chose to compare the performance on the two image representations (residual and decoded frames) with the help of two trained object detectors: YOLOv4 and tiny YOLOv4. Over the last few years, You Only Look Once (YOLO) has been one of the state-of-the-art single-stage detectors on datasets such

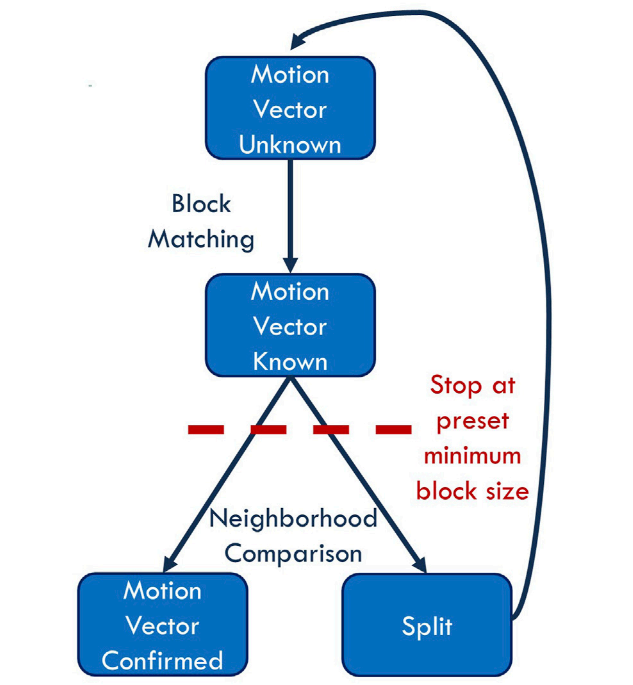

FIGURE 3 | Block diagram of the generic adaptive block matching algorithm (ABMA).

as MS-COCO (Lin et al., 2014), thanks to its different updates and versions. As the name suggests, the whole frame is scanned in a single evaluation, making the inference faster and allowing the detector to achieve real-time performance. To do so, YOLO divides the frame into pre-defined grid cells, each responsible for detecting objects thanks to YOLO's anchor boxes (prior boxes) of different shapes. Thus, each cell can produce multiple predictions containing the bounding box dimensions as well as the object class and certainty.

In its fourth revision, the authors present many improvements that make YOLO faster and more robust (Bochkovskiy et al., 2020). Among the most important ones that improve training, are CutMix and Mosaic data augmentation techniques (Yun et al., 2019), DropBlock regularization method, Cross mini-Batch Normalization (CmBN), and Self-Adversarial Training (SAT). To improve inference time, they notably introduced Mish activation function (Misra, 2019), SPP-block (He et al., 2015), Cross-stage partial connections (CSP) (Wang et al., 2020b), PAN path-aggregation block (Liu et al., 2018), and Multi-input weighted residual connections (MiWRC).

The tiny version of YOLO follows the same principles, with a drastically reduced network size. Basically, the number of convolutional layers in the backbone are scaled down, as is the number of anchor boxes that make the predictions. Consequently, it infers the result much quicker but often leads to models with lower accuracy.

One could consider applying a thresholding method to this light representation to obtain a binary image and find the contours of the objects using the algorithms presented in (Suzuki and Abe, 1985; Ren et al., 2002). Such techniques are commonly used on top of frame differencing methods. Yet, they only work in simple scenarios, since close objects are often seen as one. Deep learning-based methods are more convenient to cover more complex scenarios such as dense traffic or crowded scenes, as they are able to recognize the object shapes.

3.3 Object Trackers

The role of an object tracker is to associate the object detections with the same identities over successive frames. We chose to run two object trackers on top of our two detector models, resulting in a combination of four algorithms on both decoded and residual frames.

IOU tracker (Bochinski et al., 2017) is a very simple algorithm that relies on the assumption that detections of an object highly overlap on successive frames, resulting in a high Intersection Over Union (IOU) score. Although this is true for high refresh-rate video sequences, this is less the case for videos from traffic surveillance cameras, which run at lower refresh-rates and where vehicles travel a larger distance between two frames. To address this constraint, we chose the Kalman-IOU tracker (KIOU) instead, where a Kalman filter is added to better estimate object location and speed. This filter also lets you retain a history of the objects and re-identify them in case of missing detections. It was slightly adapted in order to work in an online tracking context, i.e., to work simultaneously with the detector.

The second tracker we chose is Simple Online and Realtime Tracking (SORT) (Bewley et al., 2016). This algorithm also includes a Kalman filter to predict existing targets' locations. It computes an assignment cost matrix between those predictions and the provided detections on the current frame using the IOU distance and then solves it optimally using the Hungarian algorithm.

Both trackers are localization-based trackers as they use only a fast statistical approach based on localization of bounding boxes. There also exists more complex trackers called feature-based trackers, such as DeepSORT (Wojke et al., 2017), which also base their predictions on objects' appearance information. However, detailed features such as vehicle models, brands, and colors cannot be distinguished in residual frames. Therefore, we limited our exploration to the former kind of trackers.

3.4 HOTA Evaluation Metric

In this section, we introduce the Higher Order Tracking Accuracy (HOTA) evaluation metric developed in detail by (Luiten et al., 2020). This metric was used to assess the performance of our detector/tracker combinations on the multi-object tracking task. In their work, they allow to measure the performance of the two stages (the detection and the association) evenly in a single metric. They also show that the HOTA metric should be used instead of the MOTA metric (Bernardin and Stiefelhagen, 2008) because the latter is biased towards detection. Ground-truth annotations and predictions are matched thanks to the Hungarian algorithm, provided that their similarity score is above a threshold α.

The HOTA score is computed by means of sub-metrics that can also be used for deeper analysis. The matching is done at the

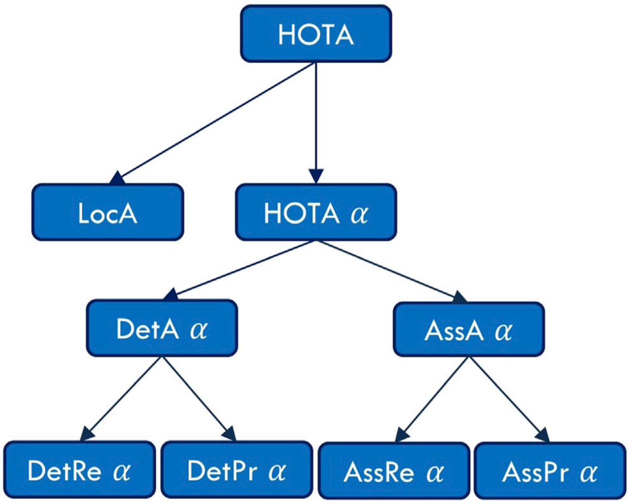

FIGURE 4 | Hierarchical diagram of the HOTA sub-metrics illustrating the different error types (Luiten et al., 2020).

detection level in each frame based on the similarity score. The matched pairs of detections are called the true positives (TP). Predictions that are not matched with a ground-truth detection are called false positives (FP). Likewise, ground-truth detections that are not matched with a prediction are called false negatives (FN). Then, the detection precision (DetPr), recall (DetRe), and accuracy (DetA) are obtained using **Eqs 1–3** respectively. More specifically, the detection recall measures the performance in finding all the ground-truth detections while the detection precision evaluates how well the predictor does not produce extra detections.

$$\mathrm{DetPr}_\alpha = \frac{|\mathrm{TP}|}{|\mathrm{TP}| + |\mathrm{FP}|} \tag{1}$$

$$\mathrm{DetPr}_\alpha = \frac{|\mathrm{TP}|}{|\mathrm{TP}| + |\mathrm{FN}|} \tag{2}$$

$$\mathrm{DetPr}_\alpha = \frac{|\mathrm{TP}|}{|\mathrm{TP}| + |\mathrm{FN}| + |\mathrm{FP}|} \tag{3}$$

The tracks' association can be computed for each matched detection. This is done by evaluating the alignment between the predicted detection's track and ground-truth detection's track. Then, the matching detections between the two tracks are called true positives (TPA). The remaining detections from the predicted track are the false positives (FPA) and the ones from the ground-truth track are the false negatives (FNA). When the best matching tracks are found, association precision (AssPr), recall (AssRe), and association (AssA) are computed using in **Eqs 4–6** respectively. This time, the recall tells how well the predictor does not split the tracks of the objects whereas the precision assesses how it avoids merging the tracks of different objects.

$$\mathrm{AssPr}_\alpha = \frac{1}{|\mathrm{TP}|} \sum_{c \in \{\mathrm{TP}\}} \frac{|\mathrm{TPA}(c)|}{|\mathrm{TPA}(c)| + |\mathrm{FPA}(c)|} \tag{4}$$

$$\mathrm{AssRe}_\alpha = \frac{1}{|TP|} \sum_{c \in \{TP\}} \frac{|\mathrm{TPA}(c)|}{|\mathrm{TPA}(c)| + |\mathrm{FNA}(c)|} \tag{5}$$

$$\mathrm{AssA}_\alpha = \frac{\mathrm{AssRe}_\alpha \cdot \mathrm{AssPr}_\alpha}{\mathrm{AssRe}_\alpha + \mathrm{AssPr}_\alpha - \mathrm{AssRe}_\alpha \cdot \mathrm{AssPr}_\alpha} \tag{6}$$

In short, HOTA_α combines the detection and the association accuracies. Each of them measures the overall quality of their own stage and can be broken down to obtain the recalls and the precisions. This is illustrated in **Figure 4**. The final HOTA score is the average of the nineteen HOTA scores computed at each similarity score threshold (ranging from 0.05 to 0.95). In the case of bounding boxes, the similarity score (S) is the Intersection Over Union (IOU). Additionally, the localization accuracy (LocA) measures the overall spatial alignment between the predicted detections and the ground-truth annotations. HOTA_α, HOTA and LocA can be calculated using **Eqs 7–9** respectively.

$$\mathrm{HOTA}_\alpha = \sqrt{\mathrm{DetA}_\alpha \cdot \mathrm{AssA}_\alpha} \tag{7}$$

$$\mathrm{HOTA} = \int_0^1 \mathrm{HOTA}_\alpha d\alpha \approx \frac{1}{19} \sum_{\alpha \in \{0.05, \ldots 0.95\}} \mathrm{HOTA}_\alpha \tag{8}$$

$$\mathrm{LocA} = \int_0^1 \frac{1}{|TP_\alpha|} \sum_{c \in \{\mathrm{TP}_\alpha\}} \mathcal{S}(c) d\alpha \tag{9}$$

3.5 Datasets

Three datasets were used to train and test our solutions. First, MIO-TCD Localization (Luo et al., 2018) allowed us to train our detectors for decoded frames. This public dataset acquired 140,000 annotated images captured at different times of the day and periods of the year by 8,000 traffic cameras deployed over North America. We trained a YOLOv4 and a tiny YOLOv4 detectors on MIO-TCD Localization and obtained an mAP of 80.4 and 71.5%, respectively, on the test set. The former is the state-of-the-art detector reported on the MIO-TCD localization challenge. Note that eleven mobile object classes were annotated in this challenge, but in this work all predictions were grouped into a single mobile object class to match the residual frames' annotations. This has also been done to ease the detection task, given that this is a first exploration into deep learning object detection using residual frames. Another reason is because an implementation to evaluate multi-class dataset with HOTA metrics is not yet available.

Second, we applied our compression algorithm on video sequences from AICity Challenge 2021 Track 1 (Naphade et al., 2021; Naphade et al., 2018) to obtain the residual frames dataset. We then trained a YOLOv4 and a tiny YOLOv4 detector on this dataset. For this purpose, we manually annotated 14,000 residual frames from six different points of view to make it appropriate for training. Indeed, there are some visibility discrepancies between decoded and residual frames. The latter representation relies on movement in the observed scene, leading to stationary vehicles often not being visible. Using an Nvidia GTX 1080Ti, it took approximately

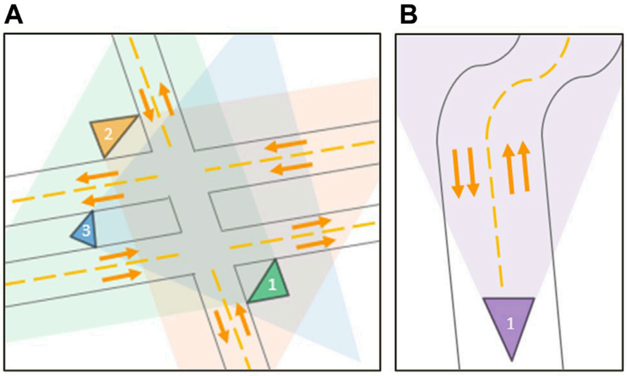

FIGURE 5 | Visualization of the two video surveillance scenarios. Experiment One **(A)** shows a complex scenario where an intersection is observed by three distinct cameras. Experiment Two **(B)** shows a simple scenario where a two-way street is observed by a single camera.

16 hours to train YOLOv4 and only 2 hours to train the tiny version on this adapted dataset.

Finally, we used AICity Challenge 2021 Track 3, also known as CityFlowV2 (Naphade et al., 2021; Tang et al., 2019). We selected four full HD (1,920 × 1,080 pixels) video sequences (recorded at 10 frames per second), resulting in a total of 10,000 frames to test the performance of the four detector/tracker combinations on both representations. CityFlowV2 provides full ground-truths with vehicle IDs. It should be noted that the challenge targets multi-camera tracking. Therefore, only objects that travel across at least two cameras were annotated. Also, vehicles whose bounding boxes were smaller than 1,000 square pixels (smaller than 0.05% of the native resolution) were not annotated. To keep a fair comparison, predictions smaller than this threshold were also removed. Otherwise, detectors would have been wrongly penalized, since they are capable of detecting further objects resulting in false positives. Therefore, the appropriate test sequences were chosen to take the aforementioned constraints into account.

3.6 Experiments

In this paper, we set up two experiments on two different video surveillance scenarios to show that an object tracker trained and tested on residual frames can be just as effective as an object tracker trained and tested on decoded frames. Both experiments followed the same setup. The two experiments were tested on four detector/tracker combinations for both image representations (residual and decoded frames): YOLOv4/KIOU, YOLOv4/SORT, tiny YOLOv4/KIOU and tiny YOLOv4/SORT.

The scenario for Experiment One is shown in **Figure 5A**. Three cameras observed the same intersection between a double-lane two-way street and a single-lane two-way street. This scenario is indeed very complex, with overlapping numbers of vehicles, and can be used for various video surveillance tasks such as traffic light violation detection, vehicle counting, and traffic jam monitoring. The goal of Experiment One was to show that a network trained and tested on residual frames could compete with a network trained and tested on decoded frames even in a highly complex scenario. The scenario for Experiment Two is shown in **Figure 5B**. One camera observed a double-lane two-way street with uninterrupted traffic flow. This scenario is mainly used for vehicle counting and wrong-way driving violation detection. The goal of Experiment Two was to highlight the benefits of using residual frames in these types of scenarios, as the constant traffic flow should allow the block matching algorithm to generate motion vectors constantly. This would lead to uninterrupted and more visible residual frames in return.

4 RESULTS

4.1 Experiment One

In this scenario, we tested the combinations on the intersection depicted in **Figure 5A**. The total footage of the three cameras amounted to 6,000 frames recorded at 10 frames per second. We calculated the HOTA metric and sub-metrics for each detector/tracker combination on both representations. The results for Experiment One are shown in **Table 1**.

For the HOTA metric, we obtained an average score of 35.92% when residual frames were the input versus an average score of 41.87% when decoded frames were the input. Concerning the DetA sub-metric, we observed an average score of 32.64% when residual frames were the input versus an average score of 34.66% when decoded frames were the input. As for the AssA sub-metric, we derived an average score of 41.21% when residual frames were the input versus an average score of 52.77% when decoded frames were the input.

On the whole, the HOTA score is 6% better on average when decoded frames are used. This difference mainly comes from AssA (11.5% difference on average) and more specifically from the association recall (AssRe) that measures how objects' tracks are split into multiple tracks. This can be explained simply by the traffic light stop lines. When vehicles stand still in the residual frames, they temporary disappear, given that residual frames depend on motion vectors generated by block matching. These vehicles are then assigned new IDs when they start moving again. On the other hand, the DetA sub-score is not much affected by residual frames (only 2% difference on average). Likewise, the detection recall (DetRe) is strongly affected by the stop lines since it measures to what extent all detections are found, but is counterbalanced by the detection precision (DetPr), which is higher for residual frames detectors thanks to the background suppression it provides. This results in fewer false positive detections. That being said, both kinds of detectors locate the objects properly in the space as shown by **Figure 6**.

4.2 Experiment Two

In this scenario, we tested the combinations on the street depicted in **Figure 5B**. The total footage of the camera amounted to 4,000 frames recorded at 10 frames per second. We calculated the HOTA metric and sub-metrics for each detector/tracker combination on both representations. The results for Experiment Two are shown in **Table 2**.

For the HOTA metric, we obtained an average score of 37.38% when residual frames were the input versus an average score of

TABLE 1 | Results of Experiment One (complex scenario where three distinct cameras observe an intersection) for residual versus decoded frames for each detector/tracker combination evaluated with HOTA metric and sub-metrics. Bold values highlight the overall HOTA scores and the underlined values show the best average scores for each metric between the two representations.

Representation	Detector/Tracker	HOTA	DetA	DetRe	DetPr	AssA	AssRe	AssPr
Residual	YOLOv4/KIOU	**36.28**	34.04	44.24	47.83	39.95	51.11	60.89
	YOLOv4/SORT	**37.47**	32.96	42.37	47.43	44.79	57.14	61.30
	Tiny YOLOv4/KIOU	**34.90**	32.67	44.73	45.31	38.55	50.64	60.26
	Tiny YOLOv4/SORT	**35.02**	30.91	41.61	43.79	41.55	54.84	57.88
Average scores		**35.92**	32.64	43.24	46.09	41.21	53.43	60.08
Decoded	YOLOv4/KIOU	**42.63**	35.97	61.26	38.53	52.94	67.74	61.37
	YOLOv4/SORT	**43.21**	35.29	59.70	38.21	55.46	69.44	61.89
	Tiny YOLOv4/KIOU	**40.35**	34.00	62.77	36.22	49.75	63.93	60.78
	Tiny YOLOv4/SORT	**41.29**	33.38	61.06	35.92	52.95	66.86	60.49
Average scores		**41.87**	34.66	61.20	37.22	52.77	66.99	61.13

TABLE 2 | Results of Experiment Two (simple scenario where a single camera observes a two-way street) for residual versus decoded frames for each detector/tracker combination evaluated with HOTA metric and sub-metrics. Bold values highlight the overall HOTA scores and the underlined values show the best average scores for each metric between the two representations.

Representation	Detector/Tracker	HOTA	DetA	DetRe	DetPr	AssA	AssRe	AssPr
Residual	YOLOv4/KIOU	**35.57**	37.96	51.11	46.60	33.60	41.15	54.92
	YOLOv4/SORT	**38.97**	37.87	50.67	46.86	40.33	50.25	55.18
	Tiny YOLOv4/KIOU	**34.92**	39.74	53.60	46.86	30.87	39.58	51.55
	Tiny YOLOv4/SORT	**40.07**	40.24	53.63	47.75	40.17	50.41	53.89
Average scores		**37.38**	38.96	52.25	47.02	36.24	45.35	53.89
Decoded	YOLOv4/KIOU	**31.72**	32.35	45.06	39.96	31.38	39.76	48.41
	YOLOv4/SORT	**33.93**	31.92	44.25	39.93	36.46	46.51	48.80
	Tiny YOLOv4/KIOU	**31.13**	27.85	44.93	32.85	34.95	41.76	49.12
	Tiny YOLOv4/SORT	**32.23**	27.69	44.09	32.93	37.69	45.29	48.78
Average scores		**32.25**	29.95	44.58	36.42	35.12	43.33	48.78

32.25% when decoded frames were the input. Concerning the DetA sub-metric, we observed an average score of 38.96% when residual frames were the input versus an average score of 29.95% when decoded frames ere the. As for the AssA sub-metric, we derived an average score of 36.24% when residual frames were the input versus an average score of 35.12% when decoded frames were the input.

Overall, the HOTA score is 5% higher on average for the residual representation in Experiment Two. This time, the main difference comes from the DetA sub-metric, where a 9% difference on average can be noticed. This is explained by a higher detection recall (DetRe) due to the vehicles' constant movements causing them to appear in the residual frames. Also, fewer false positive detection results in a better precision (DetPr), similar to Experiment One. Less significantly, the association accuracy (AssA) is quite similar for both representations, with less than 1% difference on average. Those scores are closer since tracks are not split anymore in the case of residual frames. Furthermore, the camera is closer to the ground, making the distant vehicles less distinguishable for the trackers. Consequently, all the association scores are a bit lower than in Experiment One. With everything considered, both detectors still perform generally well, as shown by **Figure 7**.

5 DISCUSSION

In this section, we first discuss the benefits and the drawbacks of using the two frame representations. Secondly, we assess both residual and decoded frames on a privacy-friendly model. Thirdly, we position our research with respect to the state-of-the-art. Last, we expose the limitations of our work.

5.1 Decoded Frames Versus Residual Frames

The results of the two experiments yield two different observations. The first observation is seen in Experiment One (complex scenario where three distinct cameras observe an intersection) where the decoded frame representation outperforms the residual frame representation. This is due to the continuously interrupted traffic flow by the intersection's traffic lights. Given that compression codecs rely on movement to generate residual frames, standstill objects do not appear in the image, as shown in **Figure 8**. Yet, residual frames are not completely limited in the scenario. For instance, it is still possible to count vehicles entering or exiting the intersection for statistical purposes.

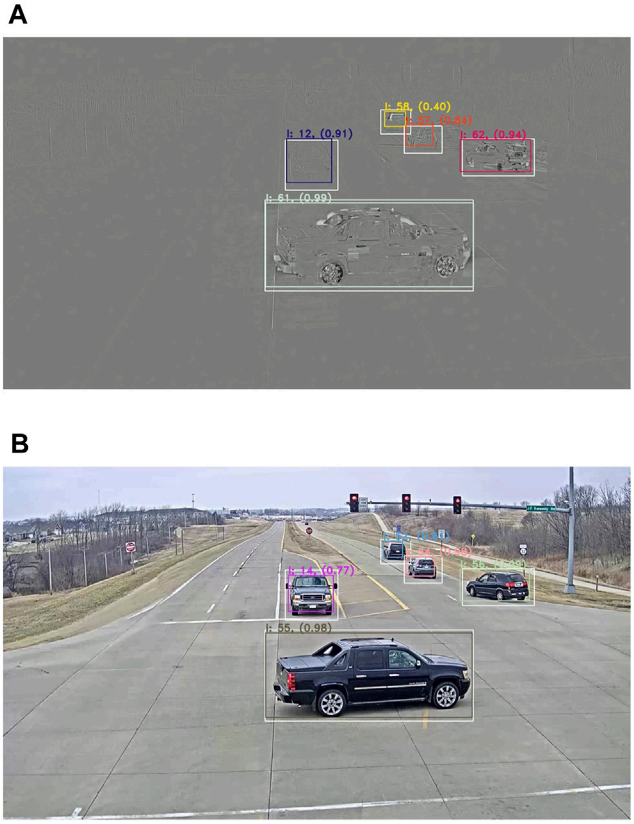

FIGURE 6 | Results on residual frames **(A)** versus decoded frames **(B)** in Experiment One. White bounding boxes represent the ground truth annotations, colored boxes represent the detections together with their respective IDs and confidence score.

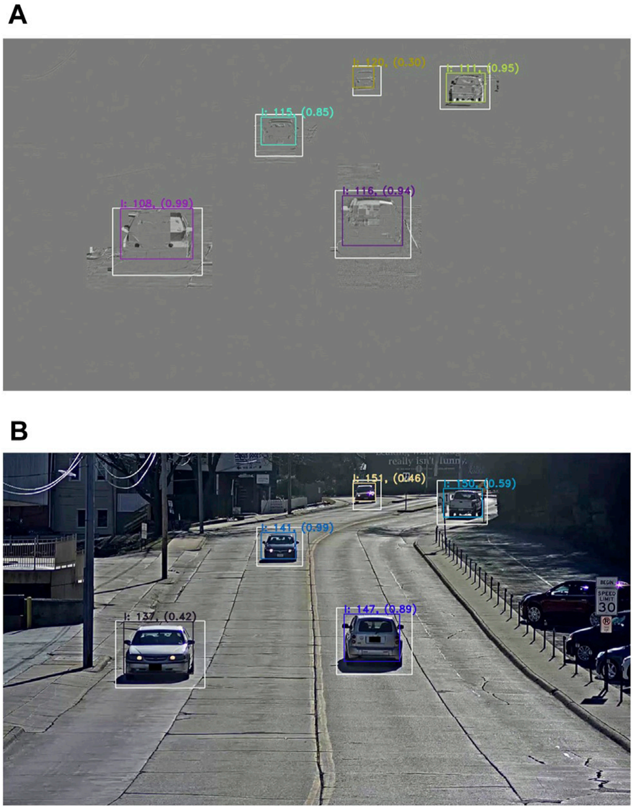

FIGURE 7 | Results on residual frames **(A)** versus decoded frames **(B)** in Experiment Two. White bounding boxes represent the ground truth annotations, colored boxes represent the detections together with their respective IDs and confidence score.

The second observation is seen in Experiment Two (simple scenario where a single camera observes a two-way street) where the residual frames representation outperforms the decoded frame representation. **Figure 9** shows a benefit and a drawback of using residual frames. On one hand, residual frames face possible detection mergers due to the uniform color distribution. On the other hand, the image smoothing offered by residual frames allows to get rid of false positives, which are sometimes predicted by deep learning techniques because of confusing shapes or colors. **Figure 10** shows a second benefit to the use of the residual frame representation. It makes it possible to deal with backgrounds that contain objects that the model is able to detect but are not of interest. While it is possible to use a mask to perform detection only in a region of interest, this is not possible in this case, where there is a full parking lot in the background.

In summary, in the complex scenario, the drawback caused by the high frequency of continuously interrupted targets outweighs the benefits of residual frames' background subtraction. However, in the simple scenario, the uninterrupted traffic flow limits the drawback of residual frames and emphasizes the benefits of its background subtraction. Consequently, for object tracking purposes, the choice of the image representation may depend on the evaluated scenario. Yet, for storage purposes, it is obvious that we would rather choose lightweight residual frames over heavy decoded frames. Also, for privacy purposes, the choice is not straightforward. It remains an open-ended question whether decoded or residual frames are more privacy-friendly. This will be discussed in the next section.

5.2 Privacy-Friendly Model

In this paper, we not only want to address the two needs of data compression and data analysis but also tackle the need for data privacy. It is, however, very difficult to define a clear evaluation metric when measuring data privacy. Also, every country has different thresholds for the amount of information that may be leaked from video surveillance footage. In the European Union, the General Data Protection Regulation (GDPR) ensures the individual's right to ask for any information held about them, including but not limited to CCTV footage[2]. It is extremely hard to guarantee with 100% accuracy that one image, regardless of the representation used, does not reveal any private information about individuals in the visual scene. In our situation, we have to compromise between effective tracking and protecting the

[2] https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016R0679&from=EN

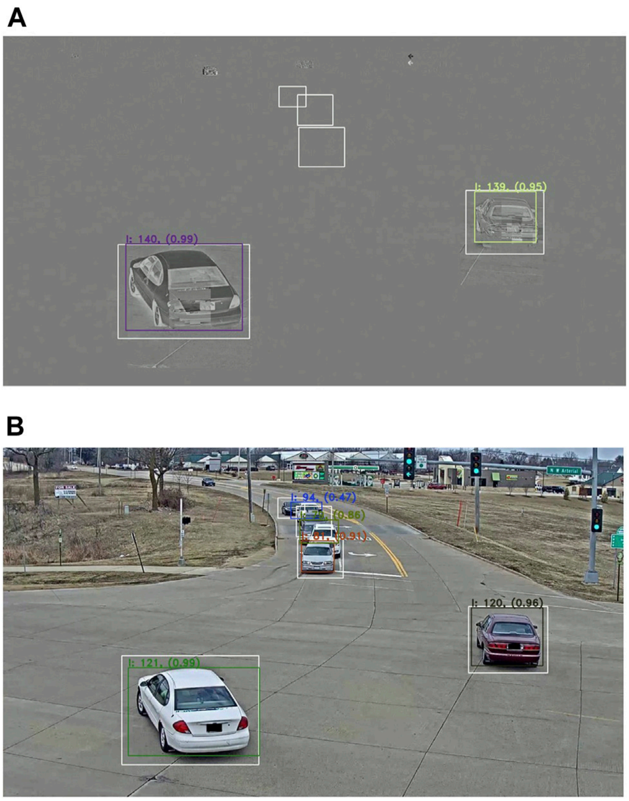

FIGURE 8 | Results on residual frames **(A)** versus decoded frames **(B)** showing false negatives in residual frames when cars are stopped. White bounding boxes represent the ground truth annotations, colored boxes represent the detections together with their respective IDs and confidence score.

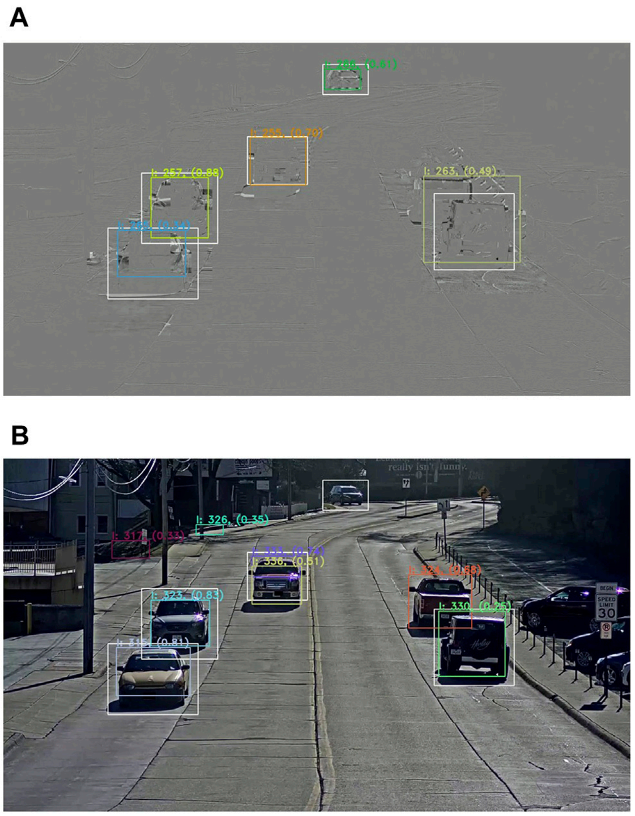

FIGURE 9 | Results on residual frames **(A)** versus decoded frames **(B)** showing false positives in the background and foreground of the decoded frame and detection mergers in the residual frame. White bounding boxes represent the ground truth annotations, colored boxes represent the detections together with their respective IDs and confidence score.

private information of individuals in the field. We can observe that the residual frames representation used is a sort of information filter on the entire image achieved by removing the background and distorting the foreground.

However, all the previously presented traditional methods of privacy modeling look at the explicit features for identification, such as visual text or facial features, only; they do not include implicit features such as location, time, and actions observed in the scene (Dufaux and Ebrahimi, 2006; Carrillo et al., 2008). To have a global privacy loss measurement, we not only need to consider all the features involved in the observed frames but should also have a non-binary evaluation metric to reflect this trade-off. To this end, a global privacy loss metric (Γ) has been put forward by Saini et al. (2010). The metric takes into consideration the four key information features that can be associated to detected objects: Who, What, When, and Where. The Who information features represent the explicit features associated with identity. The What, When and Where information features represent the implicit features that, if combined with contextual knowledge of the scene and accumulated over several frames, can represent identity with a certain level of certainty. All four key information features have scores ranging from 0 to 1, with 0 indicating no evidence of privacy loss and 1 indicating sufficient evidence of privacy loss resulting in identification. The logistic function modeling the privacy loss is shown in **Eq. 10**:

$$\Gamma = \frac{1}{1 + e^{-\left(\alpha^{*}\left(I_{Who} + I_{What,When,Where}\right) - \beta\right)}} \tag{10}$$

where α is the scaling coefficient and β is the translation coefficient. I_{Who} and $I_{What,When,Where}$ are the privacy leakage due to the explicit and implicit features, respectively. It should be noted that I_{Who} carries the highest weight of the four key information features.

In the initial model of using residual frames proposed by this paper, we would opt for encrypting the entire encoded bit-stream except for the residual frames, which would be publicly available. If we wanted to improve the privacy of our system further, we would need to apply a simple clustering filter to the residual frames. The filter would simply group together every YxZ pixel cluster by replacing them with their median value. A sample of the filtered residual frames is shown in **Figure 11** with Y = Z = 7. Thus, to apply this clustering filter, we would need to tweak the compression codec used. This means that on top of having the residual frame (needed for decoding) in the encoded bit-stream, we would have an additional filtered residual frame. In that case

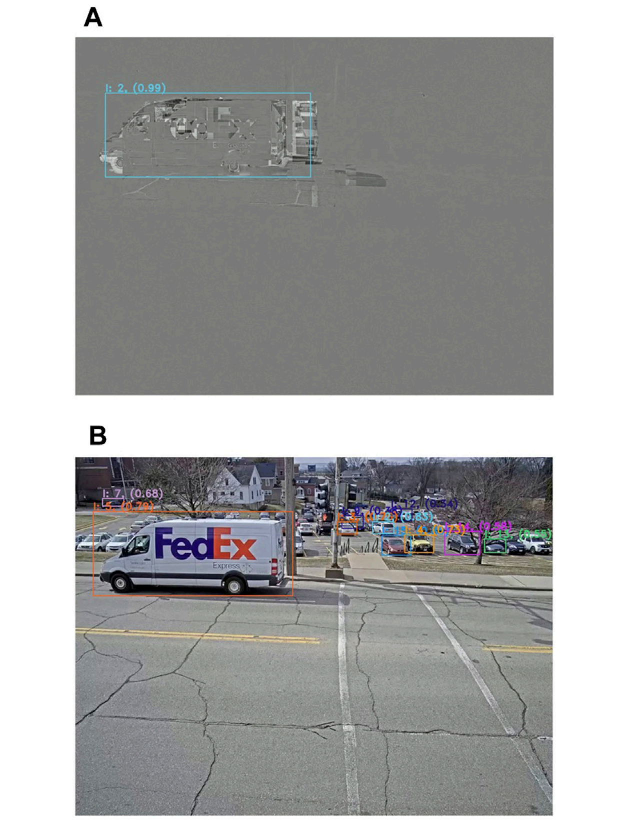

FIGURE 10 | Results on residual frames **(A)** versus decoded frames **(B)** showing a parking with irrelevant cars in the background. Colored boxes represent the detections together with their respective IDs and confidence score.

we would encrypt the residual frame alongside the rest of the stream and keep only the filtered residual frame public. However, this gain in privacy comes at a higher encoding cost for the compression codec as well as a performance drop for the deep learning object detectors and trackers. In short, there will always be a trade-off between the three needs of compression, analysis, and privacy.

We put the model into practice by enumerating the different explicit and implicit features in the case of traffic video surveillance and assigned them scores ranging from very low to very high evidence of privacy loss for each of the three proposed representations. This is depicted in **Figure 12**. Among the explicit features (I_{Who}), the evidence of privacy loss drops for colors, since residual representations highlight the prediction error, which is mapped onto a grayscale image. The brand and model of a vehicle is less recognizable compared with decoded frames, in particular for filtered residual frames. License plates, for their part, can directly leak the identity in the case of the decoded frames, but are less likely to be readable in residual frames and are scrambled in filtered residual frames. The same goes for dents and other damage, which can be considered as a unique feature on someone's vehicle. As for implicit features, I_{Where} can be decomposed mainly into text and background features. The former are texts featured on traffic signs, for instance, while the latter can be any building or monument capable of revealing the location. As previously shown, the background filtering present in the residual representations addresses this concern. I_{When} considered mainly time related information. For example, the weather and the sunlight can communicate information on the day and time of the scene. Residual representations are mostly agnostic of I_{When} features. Some exceptions could occur in the case of severe weather conditions. I_{What} can be split into simple and complex detection tasks. Simple tasks such as vehicle counting or wrong-way driving violations can be carried just as effectively with residual representations or decoded frames. Complex tasks such as detecting emergency vehicles are easier to achieve when dealing with decoded frames rather than both residual representations. Overall, we observe that the global privacy loss is better for residual frames than for decoded frames and can also be improved by applying clustering filters to the residual frames.

5.3 Research Positioning

The proposed work is a new approach to object tracking based exclusively on the analysis of compressed domain residual frames. We therefore opted to position our paper not only on its object tracking results but also by highlighting its other benefits by observing key similarities and differences with the previous works mentioned in **Section 2**. A comparison can be made with respect to the proposed

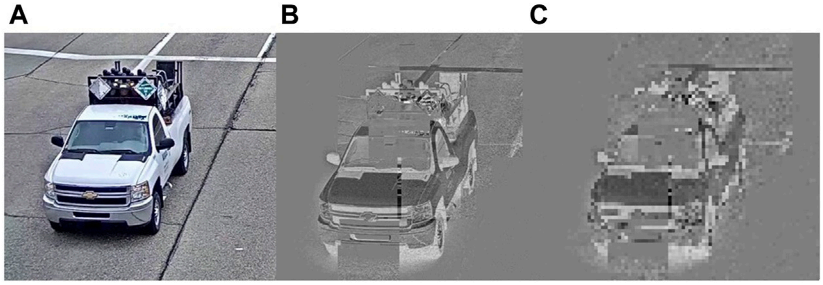

FIGURE 11 | Comparison of explicit features leakage in **(A)** decoded frames, **(B)** residual frames and **(C)** filtered residual frames.

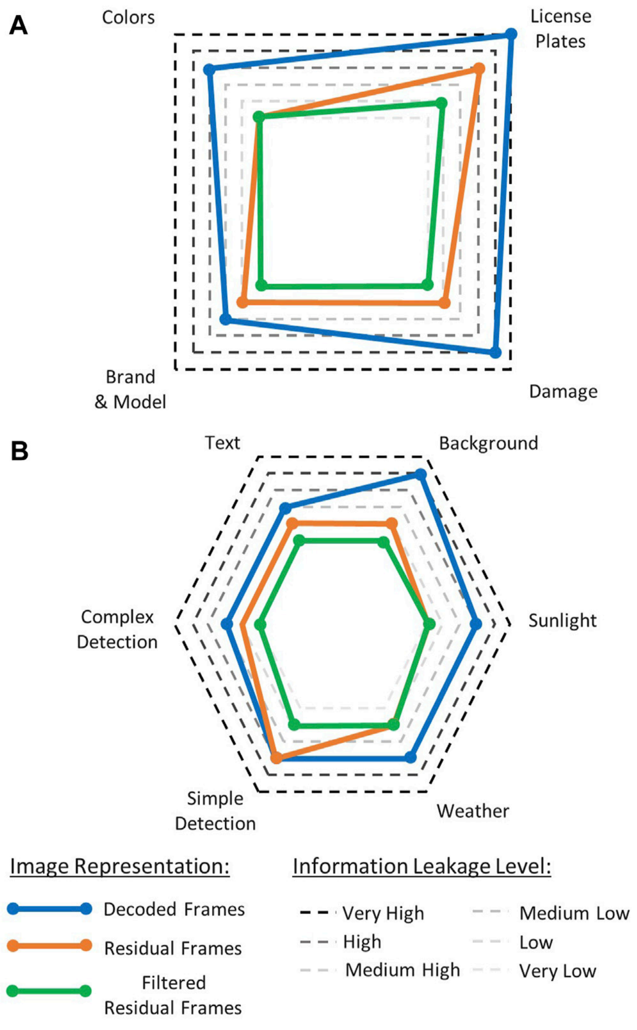

FIGURE 12 | Radar chart for both explicit **(A)** and implicit **(B)** features highlighting the reduction in information leakage when using residual and filtered residual frames over decoded frames.

work on CNN training using the ROI extracted by classical background subtraction (Kim et al., 2018). Similarly to their work, we also take advantage of the residual frame's background subtraction by-product to obtain the changing ROI to train our network. However, contrarily to their proposal, the residual frame's background subtraction by-product is auto-generated by the already existing video compression codec and therefore does not require any supplementary computational power. In the same scope, we find further similarities of our work with the VCM proposal by MPEG (Duan et al., 2020) as the two works aim to facilitate feature extraction. Another comparison can be made with regards to research propositions that have integrated compressed domain motion vectors for object detection (Ujiie et al., 2018; Liu et al., 2019). Even though we have utilized the compressed domain residual frames in our case, we still differ from their work as we propose to train and test our network exclusively on the residual frames. This will ensure that we do not rely on the original key frames (also called I frames) nor on the motion vectors for the detection and tracking process. We therefore are not only able to store our data in the compressed format but we also show that this alternative could potentially provide a privacy-friendly solution to deep learning-based object tracking. We also find similarities with the privacy-friendly video surveillance scrambler proposal that distorts the ROI in the frame to hide critical information (Dufaux and Ebrahimi, 2006). In our paper, we use the clustering filter proposed in **Section 5.2** to scramble the residual frame-generated ROI. Given that the ROI are auto-generated by the residual frames and that the clustering filtered is a basic median value filter, our algorithm would only require minor additional computational power compared to having to extract the ROI with complex algorithms.

5.4 Limitations

As for all research work, we reached some limitations that were either external or based on decisions made within our team. Firstly, we decided to chose the same parameters for all video sequences for the detectors and trackers. We decided to set parameters that worked well for all sequences because optimizing parameters for each one would have been arbitrary and could lead to biased results. For example, depending on the scenario, one could adjust a detector to favor false positives over false negatives, such as in intrusion detection systems. Conversely, urban planners would rather balance false positives and false negatives to obtain correct estimations.

Concerning the generic compression algorithm, it can be optimized in one of the two directions: either gain storage space and limit the amount of information disclosed at a cost of lower detection and tracking performance, or lose storage space and increase the amount of information leakage to improve the detection and tracking performance. In this study, we worked with fixed compression parameters for all chosen training and test sequences. The parameters have been chosen to balance storage space and tracking performance while maximizing privacy protection.

Regarding our detectors, they were trained on two different datasets. Nevertheless, they all proved to generalize well on other video sequences. Even though the detectors for residual frames were trained on AICity footage (different from those used for testing), there is less need variety in the observed scene to obtain a generic model given the simple appearance of the moving objects in the residual representation.

An important external factor that impacted our results was the AICity annotations. The fact that vehicles have to travel across at least two cameras to be annotated results in missing ground truths for vehicles passing in front of a single camera. Moreover, to ensure full coverage of the vehicles, these ground truths were annotated larger than normal. Furthermore, only annotations larger than 2/3 of the visible vehicle bodies were kept (Naphade et al., 2021;

Tang et al., 2019). All these factors do not really impact our comparison, as they are common to both kinds of detectors. However, the annotation restrictions lowered the HOTA percentages for all tested sequences.

6 CONCLUSION AND FUTURE WORK

In this work, we put forward an object tracking method which adapts both video compression and video analysis while reducing the amount of private information leakage in the video stream. This research addresses the three major needs created by the large surge in video data consumption. This was done by setting up two experiments based on two different video surveillance scenarios following the same experimental setup. The two experiments were tested on four detector/tracker combinations for both image representations (residual and decoded frames): YOLOv4/KIOU, YOLOv4/SORT, tiny YOLOv4/KIOU and tiny YOLOv4/SORT. Using the HOTA evaluation metric, we showed that using inexpensive compressed domain residual frames as an image representation can be just as effective as using decoded frames for deep learning-based object tracking. This research is to be seen as a positive result to encourage the use of compressed domain representations in deep learning-based video analysis. It is also a first step towards providing a publicly available data format for deep learning-based traffic monitoring.

Several future work propositions would extend the validation of our results. Further testing on other deep learning-based object detectors such as EfficientDet or other YOLO-family models (PP-YOLO, scaled YOLOv4, or YOLOR) should be done to consolidate our hypothesis (Tan et al., 2020; Long et al., 2020; Wang C. et al., 2020; Wang et al., 2021). Feature-based trackers such as DeepSORT, JDETracker or TPM could be trained on residual frames to see if they are capable of capturing more information, albeit at a higher cost (Peng et al., 2020; Zhang et al., 2020; Wojke et al., 2017). Another interesting path to explore is the use of residual frames in night-time object tracking, as it is much more robust to illumination and color changes than decoded frames. Finally, further exploration into the privacy evaluation metrics could be investigated with the goal of further validating our claim to providing a privacy-friendly solution to video surveillance object tracking.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/Supplementary Material, further inquiries can be directed to the corresponding author.

AUTHOR CONTRIBUTIONS

KE mainly worked on the adaptive block matching algorithm, the privacy-friendly model and put forward the proposed experimental setup. JS set up the datasets, metrics, and detection and tracking algorithms as well as conducted the experiments. BM supervised the entire project and was key in setting up the state of the art for object tracking and the privacy-friendly model. All authors contributed to the analysis of the results and discussion as well as the writing process.

FUNDING

JS is supported by the APTITUDE project (#1910045) funded by the Walloon Region Win2Wal program and ACIC S. A.

ACKNOWLEDGMENTS

We would first like to thank our colleagues at the Pixels and interactions LAB (PiLAB) for their support. We would also like to thank the participants of the (TRAIL'20) workshop for their initial contributions to the project.

REFERENCES

Barjatya, A. (2004). Block Matching Algorithms for Motion Estimation. *Final Project Paper for Spring 2004 Digital Image Processing Course at the Utah State Univ.* Available at https://www.researchgate.net/profile/Sp-Immanuel/publication/50235332"

Bernardin, K., and Stiefelhagen, R. (2008). Evaluating Multiple Object Tracking Performance: The clear Mot Metrics. *EURASIP J. Image Video Process.* 2008, 1–10. doi:10.1155/2008/246309

Bewley, A., Ge, Z., Ott, L., Ramos, F., and Upcroft, B. (2016). "Simple Online and Realtime Tracking," in 2016 IEEE International Conference on Image Processing, Phoenix, Arizona, United States, September 25–28, 2016 (ICIP), 3464–3468. doi:10.1109/ICIP.2016.7533003

Bochinski, E., Eiselein, V., and Sikora, T. (2017). "High-speed Tracking-By-Detection without Using Image Information," in 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Lecce, Italy, August 29–September 1, 2017, 1–6. doi:10.1109/avss.2017.8078516

Bochkovskiy, A., Wang, C., and Liao, H. M. (2020). *Yolov4: Optimal Speed and Accuracy of Object Detection*. CoRR abs/2004, 10934.

Carrillo, P., Kalva, H., and Magliveras, S. (2008). "Compression Independent Object Encryption for Ensuring Privacy in Video Surveillance," in 2008 IEEE International Conference on Multimedia and Expo, Hannover, Germany, June 23–26, 2008, 273–276. doi:10.1109/ICME.2008.4607424

Chien, W.-J., Zhang, L., Winken, M., Li, X., Liao, R.-L., Gao, H., et al. (2021). Motion Vector Coding and Block Merging in Versatile Video Coding Standard. *IEEE Trans. Circuits Syst. Video Tech.* 1, 3848–3861. doi:10.1109/TCSVT.2021.3101212

Duan, L., Liu, J., Yang, W., Huang, T., and Gao, W. (2020). Video Coding for Machines: A Paradigm of Collaborative Compression and Intelligent Analytics. *IEEE Trans. Image Process.* 29, 8680–8695. doi:10.1109/TIP.2020.3016485

Dufaux, F., and Ebrahimi, T. (2006). "Scrambling for Video Surveillance with Privacy," in 2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06), New York, United States, June 17–22, 2006, 160. doi:10.1109/CVPRW.2006.184

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *IEEE Trans.*

Pattern Anal. Mach. Intell. 37, 1904–1916. doi:10.1109/TPAMI.2015.2389824

Huang, Y.-W., Hsu, C.-W., Chen, C.-Y., Chuang, T.-D., Hsiang, S.-T., Chen, C.-C., et al. (2020). A Vvc Proposal with Quaternary Tree Plus Binary-Ternary Tree Coding Block Structure and Advanced Coding Techniques. *IEEE Trans. Circuits Syst. Video Tech.* 30, 1311–1325. doi:10.1109/TCSVT.2019.2945048

Kim, C., Lee, J., Han, T., and Kim, Y.-M. (2018). A Hybrid Framework Combining Background Subtraction and Deep Neural Networks for Rapid Person Detection. *J. Big Data* 5, 1–24. doi:10.1186/s40537-018-0131-x

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., et al. (2014). Microsoft Coco: Common Objects in Context. *Computer Vis. – ECCV.* 8693, 740–755. doi:10.1007/978-3-319-10602-1_48

Liu, Q., Liu, B., Wu, Y., Li, W., and Yu, N. (2019). Real-time Online Multi-Object Tracking in Compressed Domain. *IEEE Access* 7, 76489–76499. doi:10.1109/access.2019.2921975

Liu, S., Qi, L., Qin, H., Shi, J., and Jia, J. (2018). “Path Aggregation Network for Instance Segmentation,” in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, Utah, United States, June 18–22, 2018, 8759–8768. doi:10.1109/CVPR.2018.00913

Long, X., Deng, K., Wang, G., Zhang, Y., Dang, Q., Gao, Y., et al. (2020). *PP-YOLO: An Effective and Efficient Implementation of Object Detector*. CoRR abs/2007, 12099.

Luiten, J., Aljoša, O., Dendorfer, P., Torr, P., Geiger, A., Leal-Taixé, L., et al. (2020). Hota: A Higher Order Metric for Evaluating Multi-Object Tracking. *Int. J. Comp. Vis.* 129, 548–578. doi:10.1007/s11263-020-01375-2

Luo, Z., Branchaud-Charron, F., Lemaire, C., Konrad, J., Li, S., Mishra, A., et al. (2018). Mio-tcd: A New Benchmark Dataset for Vehicle Classification and Localization. *IEEE Trans. Image Process.* 27, 5129–5141. doi:10.1109/TIP.2018.2848705

Misra, D. (2019). *Mish: A Self Regularized Non-monotonic Neural Activation Function*. CoRR abs/1908, 08681.

Mukherjee, D., Bankoski, J., Grange, A., Han, J., Koleszar, J., Wilkins, P., et al. (2013). “The Latest Open-Source Video Codec Vp9 - an Overview and Preliminary Results,” in 2013 Picture Coding Symposium, San Jose, California, December 8–11, 2013 (PCS), 390–393. doi:10.1109/PCS.2013.6737765

Naphade, M., Chang, M.-C., Sharma, A., Anastasiu, D. C., Jagarlamudi, V., Chakraborty, P., et al. (2018). “The 2018 Nvidia Ai City challenge,” in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Salt Lake City, Utah, United States, June 18–22, 2018, 53–537. doi:10.1109/CVPRW.2018.00015

Naphade, M., Wang, S., Anastasiu, D. C., Tang, Z., Chang, M.-C., Yang, X., et al. (2021). “The 5th Ai City challenge,” in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Virtual, June 19–25, 2021. doi:10.1109/cvprw53098.2021.00482

Peng, J., Wang, T., Lin, W., Wang, J., See, J., Wen, S., et al. (2020). Tpm: Multiple Object Tracking with Tracklet-Plane Matching. *Pattern Recognition* 107, 107480. doi:10.1016/j.patcog.2020.107480

Ren, M., Yang, J., and Sun, H. (2002). Tracing Boundary Contours in a Binary Image. *Image Vis. Comput.* 20, 125–131. doi:10.1016/s0262-8856(01)00091-9

Saini, M., Atrey, P. K., Mehrotra, S., Emmanuel, S., and Kankanhalli, M. (2010). “Privacy Modeling for Video Data Publication,” in 2010 IEEE International Conference on Multimedia and Expo, Singapore, July 19–23, 2010, 60–65. doi:10.1109/ICME.2010.5583334

Sullivan, G. J., Ohm, J.-R., Han, W.-J., and Wiegand, T. (2012). Overview of the High Efficiency Video Coding (Hevc) Standard. *IEEE Trans. Circuits Syst. Video Tech.* 22, 1649–1668. doi:10.1109/TCSVT.2012.2221191

Suzuki, S., and Abe, K. (1985). Topological Structural Analysis of Digitized Binary Images by Border Following. *Comp. Vis. Graphics, Image Process.* 30, 32–46. doi:10.1016/0734-189X(85)90016-7

Tan, M., Pang, R., and Le, Q. V. (2020). “Efficientdet: Scalable and Efficient Object Detection,” in 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Virtual, July 14–19, 2020, 10778–10787. doi:10.1109/CVPR42600.2020.01079

Tang, Z., Naphade, M., Liu, M.-Y., Yang, X., Birchfield, S., Wang, S., et al. (2019). “Cityflow: A City-Scale Benchmark for Multi-Target Multi-Camera Vehicle Tracking and Re-identification,” in In 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, California, USA, June 16–20, 2019, 8789–8798. doi:10.1109/CVPR.2019.00900

Ujiie, T., Hiromoto, M., and Sato, T. (2018). “Interpolation-based Object Detection Using Motion Vectors for Embedded Real-Time Tracking Systems,” in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Salt Lake City, Utah, USA, June 18–22, 2018. doi:10.1109/cvprw.2018.00104

Vermaut, F., Deville, Y., Marichal, X., and Macq, B. (2001). A Distributed Adaptive Block Matching Algorithm: Dis-Abma. *Signal. Processing: Image Commun.* 16, 431–444. doi:10.1016/s0923-5965(00)00008-4

Wang, C.-Y., Mark Liao, H.-Y., Wu, Y.-H., Chen, P.-Y., Hsieh, J.-W., and Yeh, I.-H. (2020b). “Cspnet: A New Backbone that Can Enhance Learning Capability of Cnn,” in 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Virtual, June 14–19, 2020, 1571–1580. doi:10.1109/CVPRW50498.2020.00203

Wang, C., Bochkovskiy, A., and Liao, H. M. (2020a). *Scaled-yolov4: Scaling Cross Stage Partial Network*. CoRR abs/2011, 08036.

Wang, C., Yeh, I., and Liao, H. M. (2021). *You Only Learn One Representation: Unified Network for Multiple tasks*. CoRR Abs/2105, 04206.

Wojke, N., Bewley, A., and Paulus, D. (2017). “Simple Online and Realtime Tracking with a Deep Association Metric,” in 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, September 17–20, 2017 (IEEE), 3645–3649. doi:10.1109/ICIP.2017.8296962

Yun, S., Han, D., Chun, S., Oh, S. J., Yoo, Y., and Choe, J. (2019). “Cutmix: Regularization Strategy to Train strong Classifiers with Localizable Features,” in 2019 IEEE/CVF International Conference on Computer Vision, Seoul, October 27–November 2, 2019 (ICCV), 6022–6031. doi:10.1109/ICCV.2019.00612

Zhang, L., Zhang, K., Liu, H., Chuang, H. C., Wang, Y., Xu, J., et al. (2019). “History-based Motion Vector Prediction in Versatile Video Coding,” in 2019 Data Compression Conference, Snowbird, Utah, USA, March 26–29, 2019 (DCC), 43–52. doi:10.1109/DCC.2019.00012

Zhang, Y., Wang, C., Wang, X., Zeng, W., and Liu, W. (2020). *A Simple Baseline for Multi-Object tracking*. CoRR Abs/2004, 01888.

Conflict of Interest: This study received funding from ACIC S.A. The funder was not involved in the study design, collection, analysis, interpretation of data, the writing of this article or the decision to submit it for publication. All authors declare no other competing interests.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.