cryptography

MDPI

Article

Privacy Preserved Video Summarization of Road Traffic Events for IoT Smart Cities

Mehwish Tahir [1,*], Yuansong Qiao [1], Nadia Kanwal [2], Brian Lee [1] and Mamoona Naveed Asghar [3,*]

[1] Software Research Institute, Technological University of the Shannon (TUS): Midlands Midwest (Athlone Campus), N37 HD68 Athlone, Ireland
[2] School of Computer Science and Mathematics, University of Keele, Keele ST5 5BG, UK
[3] School of Computer Science, University of Galway, H91 TK33 Galway, Ireland
* Correspondence: m.tahir@research.ait.ie (M.T.); mamoona.asghar@universityofgalway.ie (M.N.A.)

Abstract: The purpose of smart surveillance systems for automatic detection of road traffic accidents is to quickly respond to minimize human and financial losses in smart cities. However, along with the self-evident benefits of surveillance applications, privacy protection remains crucial under any circumstances. Hence, to ensure the privacy of sensitive data, European General Data Protection Regulation (EU-GDPR) has come into force. EU-GDPR suggests data minimisation and data protection by design for data collection and storage. Therefore, for a privacy-aware surveillance system, this paper targets the identification of two areas of concern: (1) detection of road traffic events (accidents), and (2) privacy preserved video summarization for the detected events in the surveillance videos. The focus of this research is to categorise the traffic events for summarization of the video content, therefore, a state-of-the-art object detection algorithm, i.e., You Only Look Once (YOLO*v*5), has been employed. YOLO*v*5 is trained using a customised synthetic dataset of 600 annotated accident and non-accident video frames. Privacy preservation is achieved in two steps, firstly, a synthetic dataset is used for training and validation purposes, while, testing is performed on real-time data with an accuracy from 55% to 85%. Secondly, the real-time summarized videos (reduced video duration to 42.97% on average) are extracted and stored in an encrypted format to avoid un-trusted access to sensitive event-based data. Fernet, a symmetric encryption algorithm is applied to the summarized videos along with Diffie–Hellman (DH) key exchange algorithm and *SHA256* hash algorithm. The encryption key is deleted immediately after the encryption process, and the decryption key is generated at the system of authorised stakeholders, which prevents the key from a man-in-the-middle (MITM) attack.

Keywords: classification; cryptography; smart cities; traffic events; video summarization; YOLO

Citation: Tahir, M.; Qiao, Y.; Kanwal, N.; Lee, B.; Asghar, M.N. Privacy Preserved Video Summarization of Road Traffic Events for IoT Smart Cities. *Cryptography* **2023**, *7*, 7. https://doi.org/10.3390/cryptography7010007

Academic Editors: Hui Zhu and Yuan Zhang

Received: 20 November 2022
Revised: 25 January 2023
Accepted: 6 February 2023
Published: 9 February 2023

1. Introduction

Managing road traffic flow is one of the biggest challenges for smart cities—a subset of the Internet of Things (IoT) infrastructure. Thanks to smart surveillance applications [1] for presenting a number of practical solutions for detecting road traffic events. The extensive installation of closed-circuit television (CCTV) cameras in smart cities has changed the infrastructure of urban areas for effective monitoring and safety of people in addition to traffic on roads. Artificial intelligence (AI) is gaining traction in the area of object detection from CCTV video recordings for video analytics. As a result, people are getting accustomed to AI-based intelligent surveillance systems. CCTV cameras installed in smart cities can easily be used to detect an event like an accident, robbery, assault, etc. If these mentioned events are detected in time, swift action can be taken by the relevant service authorities. There are two significant research challenges for surveillance data, i.e., event identification and privacy preservation of visual data.

For event identification, computer vision and image processing play a vital role in computer technology for detecting and classifying objects. Numerous algorithms are pro-

posed by researchers in the past to detect and classify objects from images and videos. The most popular among object detection and classification algorithms include convolutional neural network (CNN) [2], region-based CNN (R-CNN) [3], Fast R-CNN [4], Faster R-CNN [5,6], and YOLO [7]. This paper has targeted You Only Look Once (YOLO), a deep learning model [8] for the detection of road traffic events such as accidents (in our use-case). YOLO is basically an object detection algorithm and researchers have already utilised it for several applications including abandoned baggage detection [9], mask detection [10], license plate recognition [11], shuttlecock detection [12], pothole detection [13], and many more to mention.

While dealing with visual data, another challenge is to ensure the privacy of sensitive data in recorded videos. European General Data Protection Regulation (EU-GDPR) [14] recommends data minimisation and data protection by design for all types of collected data. Therefore, besides traffic events detection from the recorded data, this study also proposed a solution for minimal data storage. For this purpose, a short video (summary) for the detected traffic events (accident only) was generated and stored in a secure manner. EU-GDPR principle of "data minimisation" (Article 5) specifies that a data controller should limit the collection of information to what is directly relevant and necessary to accomplish a certain task, while "data protection by design" (Article 25) states that embedding data privacy features and technologies directly into the design of the project ensures better cost-effective protection for data privacy. The research goals of reduced and also secured video storage through an encrypted video summarization approach are aligned with the data minimisation and data protection by design criteria of the EU-GDPR. Moreover, limited data storage will also help in the quick retrieval of desired traffic event (accident) data (from massive stored videos) for required future video analytics and legal investigations.

Deep learning models require a huge amount of data for training, which is a major challenge due to the scarcity of publicly available real-world datasets. To overcome this issue, this research study has utilised synthetic data for the training of a supervised learning model, i.e., YOLO. As an added benefit, synthetic data have a privacy-related advantage over real-world data. Finally, the summarized data needs to be protected before storing them on a device. EU-GDPR recommends only encryption as a reversible data protection safeguard [15,16]. In order to ensure GDPR compliance, we have implemented encryption to ensure that summarized videos cannot be accessed by unauthorised stakeholders. A symmetric key, naive encryption [17] was implemented for securing stored videos.

1.1. Research Questions

To make a GDPR-compliant technological solution for smart infrastructures (cities and surveillance systems), the following research questions are addressed in this paper:

RQ1: Can an object detection model trained on synthetic data (e.g., YOLO) produce optimum event detection results on real-time captured CCTV footage?
RQ2: For an event alert, can a CCTV video segment be extracted and stored in an encrypted format with foolproof security of encryption/decryption keys?

1.2. Research Contributions

By considering the aforementioned questions, the research contributions are:

RC1: YOLO is trained for event detection (rather than object detection) using a customised synthetic video dataset (accident and non-accident video frames).
RC2: The testing of the trained model is performed on real-time CCTV footage to classify the accidental and non-accidental video frames with different environmental conditions such as videos recorded at night and during rain.
RC3: Event-based video summaries of CCTV footage are stored in an encrypted format using Diffie–Hellman key exchange and Fernet cipher. The *SHA256* hash is applied to tackle key and data-level security.
RC4: An annotated customised synthetic dataset (accident and non-accident video frames) for YOLO*v*5 model training is provided for future researchers.

The remainder of this research paper is organised as follows: Section 2 sheds a light on related work, Section 3 elaborates on methodology, results and discussion is given in Section 4, Section 5 shows comparative analysis with existing techniques, Section 6 explains limitations with some future work, and in the last, Section 7 concludes this research.

2. Related Work

This section discusses related work from the literature for video summarization through YOLO and cryptography for visual data security along with the key management techniques.

2.1. Video Summarization through YOLO

In the existing literature, YOLO is mostly used to summarize sports videos. An automated sports video summarization technique presented by [18], in which YOLO was trained on 1300 images of different scores. A frame was given as input to YOLO after every second, if scoreboard was present, then score was inputted to optical character recognition (OCR) to compare the previous and current score. The model then read the score after every second and found the difference between the scores. The timestamps of important event were returned as output and then event from the video was summarized. Hence, it was a video summarization technique which read text from frames rather than detecting events based on the movement of objects/individuals. Another technique proposed by [19], which highlighted a query based object detection. Firstly, YOLO was trained to detect objects from surveillance videos and then if the required object was present in the video frames, a video summary was created. This scheme summarized videos based on objects only, whereas events and privacy of objects was not considered. An object of interest (OoI) based video summarization was proposed by [20], which also used YOLO to detect OoI and compared the object given as input to the objects in dictionary. The frames that had the required objects were saved in a summarized video. Ref. [21] also proposed a technique to create summary of video based on objects. The objects in the extracted frames were detected through YOLO and if an object was present in the video frame, it was made the part of video summary, otherwise, discarded.

The related work shows that YOLO is primarily used for object detection by researchers and the video summarization is mostly based on objects and human activities rather than on events. However, the focus of the proposed research is the utilisation of YOLO for the detection of road traffic events (accidents and non-accidents).

2.2. Cryptography for Visual Data Security

The literature survey shows many research studies on lightweight encryption techniques for IoT applications. The authors [22] proposed a lightweight selective encryption (SE) based on the region of interest (ROI), as whole video encryption is time-consuming. The tiles concepts in high efficiency video coding (HEVC) and SE for sensitive parts in the video were used to secure the videos. A chaos-based stream cipher-dependent SE encrypted a set of HEVC syntax elements that could be decoded by an HEVC decoder, but the secret key was needed when ROI had to be decrypted. A skin color-based privacy protection technique was proposed by [23], which encrypted the video by detecting skin color in real-time surveillance systems. A lightweight SE using ChaCha20 algorithm was applied for securing the videos by only encrypting key points detected through the FAST feature detector [24]. Another research proposed a solution to protect sensitive videos during streaming [25], in which chaotic-based encryption was applied and for that purpose 1-D and 2-D models were utilised. The confusion and diffusion properties of that logistic map and Arnold's cat map-based algorithm were good. The logistic map introduced a specific external key that replaced and recovered the pixels value when encryption and decryption were applied, in contrast to Arnold's cat map, which changed the dataset into a pseudo-random state over a number of repetitions and was also reversible. The database with faces was created and stored on the cloud by applying Fernet encryption for security

by [26]. The decryption key was stored on the client side and that protected the data on the cloud as data stored on the cloud could only be decrypted by the key stored on the client side. The face images stored on clouds were used to identify known and unknown faces from videos, which were then further used to extract key information from the video files through recognised faces.

Diffie–Hellman (DH) key exchange algorithm to secure the video surveillance system was applied by [27]. The elements named as secret and timestamp from DH key exchange algorithm were added to protect the surveillance system from the man-in-the-middle (MITM) attack.

The related work shows that the researchers have used lightweight encryption for SE (on full videos) to reduce the computational time consumed by encryption and decryption in real-time applications [28]. However, the focus of the proposed research is to develop a privacy preserved event-based road traffic event (accident) detection system with naive encryption of identified event-based video frames (summary). This research paper proposed Fernet encryption in securing the event-based summarized videos along with the secure key management scheme using DH key exchange algorithm and *SHA256* hash algorithm.

3. Methodology

Synthetic data is data generated artificially to appear like real-time data. The importance of synthetic data cannot be denied as a large number of training datasets can artificially be generated which is hard to collect in real-time scenarios. Furthermore, the privacy of recorded individuals/objects is compromised when datasets are made publicly available. Hence, the use of synthetic data is appreciable in privacy preservation. Figure 1 shows two synthetic data video frames with non-accident and accident road events.

(**a**) (**b**)

Figure 1. Synthetic video data frames with (**a**) non-accident, and (**b**) accident road event.

The proposed methodology shown in Figure 2 is based on training of YOLO on a customised dataset (synthetic video frames). The most stable version of YOLO, i.e., YOLO*v*5 is used for the experiments. There are four main phases in the methodology of the proposed research to implement the privacy preserved video summarization: (1) dataset preparation, (2) YOLO training, (3) testing, and (4) storage and retrieval. Below is the elaboration of the phases included in the methodology:

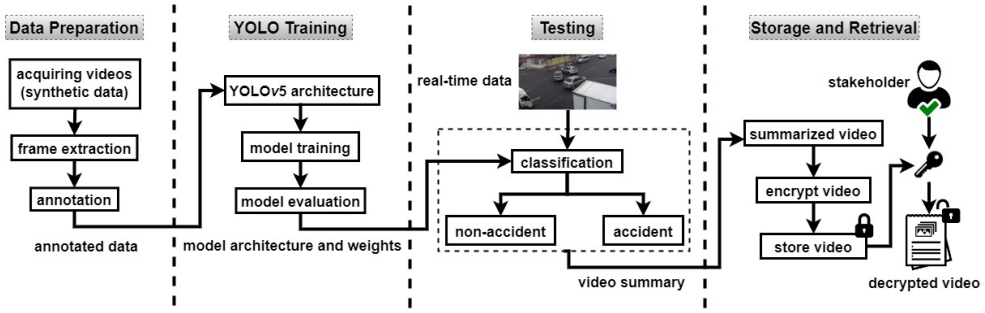


Figure 2. Proposed methodology for privacy preserved video summarization.

3.1. Dataset Preparation

The dataset preparation includes further three steps: (1) acquiring videos, (2) frame extraction, and (3) annotation. The description is as follows:

1. For the training of YOLO, high-quality road traffic videos (synthetic dataset) were taken from BeamNG drive [29].
2. YOLO required annotated data for training, therefore, frames were extracted from videos for applying annotation.
3. In the last step of the dataset preparation phase, the annotation was applied to every selected frame. Total 600 frames were selected and manually annotated by the tool [30]. The bounding boxes for accident and non-accident regions were annotated in frames as 0 and 1, respectively, as shown in Figure 3. After annotation, the frames along with their respective text files were saved on the computer drive. The text files contain the annotation information, i.e., class name and coordinates for each bounding box. After this, the YOLO was trained over the customised annotated dataset. The annotated dataset is made publicly available on Kaggle repository [31] for future researchers.

Figure 3. Customised (annotated) synthetic data used in training of YOLO*v*5 (bounding boxes drawn on event occurred: 0 (red color) and 1 (pink color) represents accident and non-accident, respectively).

3.2. YOLO Training

YOLO sees the whole frame at once making it faster to detect objects in real-time scenarios. During the training phase, YOLO is trained on the customised (annotated) synthetic dataset. This process is completed in three steps: (1) YOLO*v*5 architecture, (2) model training, and (3) model evaluation. The details of each step are given below:

1. YOLO*v*5 is pre-trained model on the COCO dataset [32]. In this research, YOLO*v*5 is trained on customised synthetic dataset, as this version is lighter and faster than previous versions. It can easily be implemented on custom datasets by making necessary modifications for the classification task at hand. As this research entailed the classification of only two classes, therefore, we re-trained it on our dataset with the existing base layers. The head of YOLO*v*5 consists of three convolutional layers that predict the location of bounding boxes (x, y, height, width) where an event has occurred,

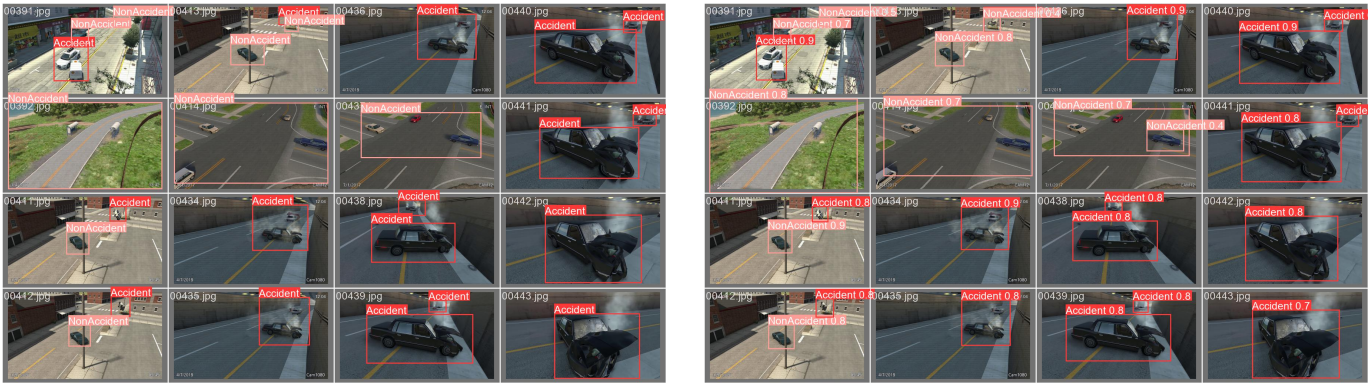
scores (certainty of the predicted event), and class of the event. YOLO$v5$ uses sigmoid linear unit (SILU) and sigmoid activation functions. SILU is also known as swish activation function and it is used in the hidden layers with convolutional operations. On the other hand, sigmoid is used in the output layer with convolutional operations.

2. YOLO$v5$ model training was done with Google Colab [33]. The dataset was split into training (80%) and validation (20%) data. The model summary comprised of number of layers, parameters, gradients, and GFLOPs is shown in Table 1. YOLO keeps the aspect ratio of the images, therefore, all the training images (1920 × 1080 resolution) were resized (416 × 234 resolution) for use in a 416 × 416 network. It took 3.939 h to train the model for 30 epochs with batch size 16. Initially, the training started on pre-trained weights, and on completion of training, two new weight files were created with the names *best.pt* and *last.pt*. We selected the file containing the best weights, i.e., *best.pt*, to test real-time videos, whereas *last.pt* holds the weights for last epoch in the training. The training results for one batch are shown in Figure 3.

Table 1. Model summary.

Layers	Parameters	Gradients	GFLOPs
157	7,015,519	0	15.8

3. The model was evaluated on synthetic data during training phase. Sample results on the validation data along with ground truth labels are shown in Figure 4a and predicted classes in validation phase are shown in Figure 4b, which shows model has learned good enough from the training dataset. However, testing is performed on real-time videos footage rather than synthetic dataset which is discussed in Section 3.3.



(**a**) (**b**)

Figure 4. (**a**) Validation data with labelled classes (ground truth), and (**b**) prediction of classes during validation (bounding boxes drawn on event occurred: accident and non-accident representation with red and pink colors, respectively).

Confusion matrix shown in Figure 5 also represents good training on customised synthetic dataset with 98% correct prediction for accidents and 95% prediction for non-accidents.

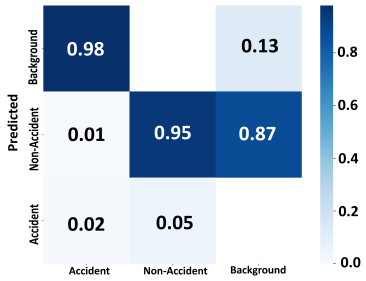


Figure 5. Confusion matix for the model training.

The model training was also evaluated through precision recall (PR) curve and F1-confidence curve. The details of both curves are as follows:

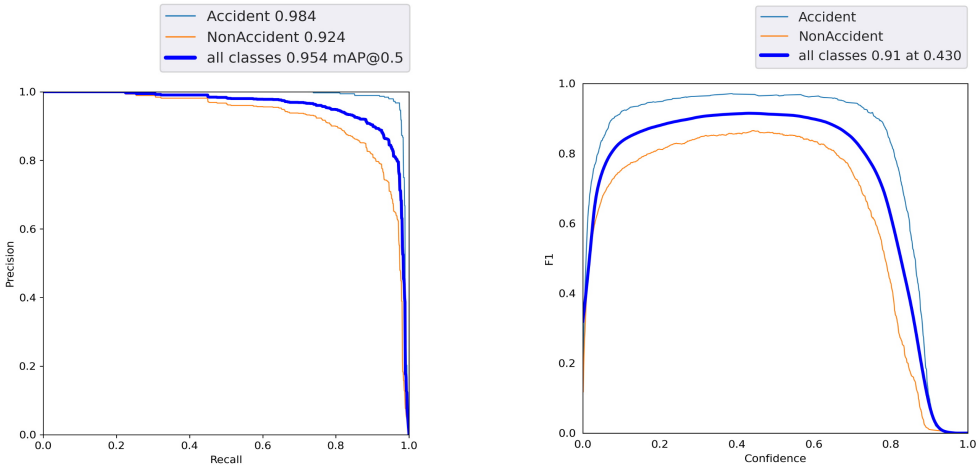
PR curve shown in Figure 6a represents that model can predict accident and non-accident classes with a score of 0.984 and 0.924, respectively, with *mAP@0.5*, where *mAP* is mean average precision threshold. The graphs contain precision (P) values on y-axis and recall (R) values on x-axis. Equations (1) and (2) were used for calculation of P and R values, where: TP is true positive, FN is false negative, and FP is false positive.

$$P = \frac{TP}{TP + FN} \tag{1}$$

$$R = \frac{TP}{TP + FP} \tag{2}$$

F1-Confidence Curve is a measure of P and R values at any specific threshold value, i.e., 0.91 here for both classes at 0.430, as shown in Figure 6b. $F1$-confidence curve value is near 1, which shows model is trained well. $F1$ score is calculated as:

$$F1\ score = 2 \times \frac{P \times R}{P + R} \tag{3}$$



(**a**) (**b**)

Figure 6. (**a**) Precision recall, and (**b**) F1-confidence curve after training on customised synthetic dataset (accident and non-accident video frames).

Results of the trained model are given in Table 2. There were total 600 images used in the training, having total 985 instances (398 for accidents and 587 for non-accidents). P value for accidents is 0.967 and 0.838 for non-accidents, on the other hand, R value is 0.971 and 0.882 for accidents and non-accidents, respectively. Overall P value is 0.903 and R value is 0.927. The mAP at threshold 50 is 0.984 for accidents, 0.924 for non-accidents, and overall, it is 0.954. The mAP value from threshold 50 to 95 is 0.704 for accidents, 0.584 for

non-accidents, and overall it is 0.646. The result shows that model has achieved sufficient level of feature learning on customised synthetic data.

Table 2. Results of the trained model on customised (annotated) synthetic dataset.

Class	Instances	P	R	mAP50	mAP50-95
All	985	0.903	0.927	0.954	0.645
Accident	398	0.967	0.971	0.984	0.704
Non-Accident	587	0.838	0.882	0.924	0.586

3.3. Testing on Real-Time Videos Footage

The trained YOLO*v*5 model was tested on real-time road accident videos footage [34]. The testing is performed on 15 different videos footage. The details of the videos footage along with average time taken by each video frame are given in Table 3, that includes video no, frames, length (sec), and, pre-process, inference, and post-process time in ms (millisecond). The pre-processing converted an image from Numpy n-dimensional arrays to Pytorch tensors and normalized the pixel values from 0 to 255 to 0.0–1.0. Time spent inside the model is known as the inference time. Similarly, time taken by non-maximum suppression (NMS) is called post-processing. The model can predict many bounding boxes, all having different positions, sizes, and confidence levels, and there is possibility of overlapping when multiple bounding boxes are predicted. Therefore, NMS keeps bounding boxes with more confidence values and discard the other overlapping bounding boxes. The sample original frames along with the frames after event detection through YOLO is shown in Figure 7. The results show that all video frames with events are successfully classified as accident and non-accidents.

Table 3. Test speed per image for real-time test videos footage.

Video No.	Frames	Length (sec)	Speed Per Image (ms)		
			Pre-Process	Inference	Post-Process
1	320	10	0.4	9.4	1.1
2	238	7	0.4	9.6	0.9
3	480	16	0.3	9.0	1.4
4	480	16	0.5	9.3	0.8
5	360	12	0.3	9.1	1.2
6	360	12	0.4	9.5	1.2
7	360	12	0.3	9.4	1.1
8	600	20	0.3	9.4	1.3
9	240	8	0.6	9.1	1.8
10	376	12	0.4	9.4	1.2
11	300	10	0.3	9.4	1.1
12	270	9	0.4	9.8	1.2
13	480	16	0.4	9.4	1.2
14	361	12	0.5	9.4	1.2
15	300	10	0.4	9.4	1.1

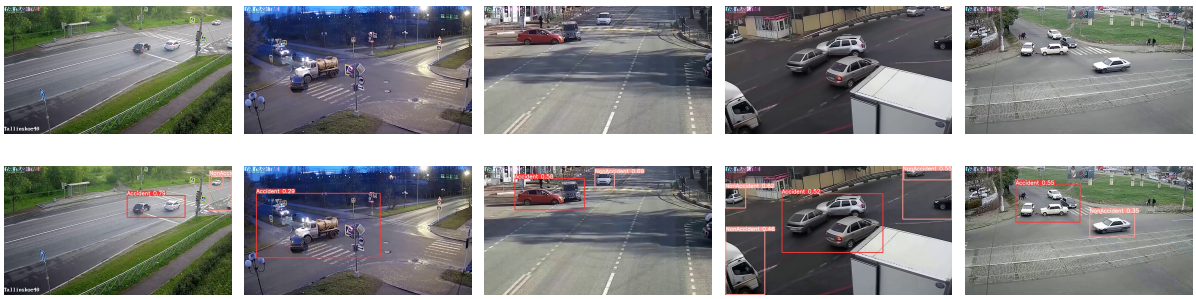


Figure 7. Original frames (top row) and event detection (bottom row with bounding boxes drawn on event occurred: accident and non-accident representation with red and pink colors, respectively) through YOLO in real-time videos footage.

3.4. Storage and Retrieval

After event detection, real-time test videos footage were summarized. All frames with detected accidents were collected for making a video summary. In every second, if an accident was detected in 50% or more of the frames, then all the frames of that second were kept for the storage. Otherwise, frames were discarded. This threshold was set to provide a context to the detected event (accident) in the video summary. For example, if a video was of 30 FPS (frames per second) and an accident was detected in 20 frames only, the remaining 10 frames were also saved with the accidental frames to show a few frames before or after the accident.

Cryptography for Summarized Videos

For securing data, naive symmetric encryption was applied to the summarized videos and stored on the server. A lightweight symmetric encryption algorithm, Fernet [35,36] was implemented to encrypt video summaries in the proposed research. Fernet uses Advanced Encryption Standard (AES) [26], i.e., Cipher Block Chaining (CBC) with a 128-bit key for encryption and a secure hash algorithm (*SHA256*) for authentication of data. The secure way of key management (generation and exchange) is a big challenge for robust encryption, therefore, Diffie–Hellman (DH) [37] key exchange method was applied to generate password keys that were run through a key derivative function and should be salted to generate encryption/decryption keys for the parties. To increase the security of a password key that has been hashed, the process of salting involves adding random characters to the password key that have a specific length. DH can be used on unsecured channels for exchanging encryption keys without knowing each other. The safety of encryption keys is mandatory, so we proposed using a hardware wallet [38] to store the DH-generated password keys. Keys can be further securely handled through key management algorithms [39], which could be taken as future work for this research.

The basic structure of Fernet encryption is XORing (exclusive-OR) data with an initialization vector (IV) followed by encryption with Fernet generated key. The decryption process decrypts the encrypted data by XORing it with an IV followed by decryption with stored Fernet generated key. Algorithm 1 presents the pseudo-code for implementing cipher on video summaries. The steps included in encrypting a summarized video through Fernet encryption, DH, and *SHA256* are also presented in Figure 8.

1. Firstly, two password keys (k_s and k_r) were generated by DH key exchange algorithm.
2. A hash value (h) was calculated for *camera ID* and stored as a digest in $salt$.
3. The password key k_s was used along with $salt$ to derive an encryption key (EK), whereas k_r was stored in a hardware wallet for generating the decryption key (DK) at the time of decryption.
4. EK was further used for preserving the privacy of a summarized video.
5. Fully encrypted video or token was stored on the server.
6. EK was deleted.

Algorithm 1: Pseudo-code for applied cryptography on summarized videos

```
/* Calculate password keys for encryption and decryption key
   generation                                                    */
Initialization: p ← any_random_number, g ← any_random_number;
Output: k_s, k_r;

/* Calculate password keys k_s and k_r through Diffie-Hellman key
   exchange algorithm                                            */
s ← select any random number;
r ← select any random number;
x ← calculate exchange key (g^s mod p);
y ← calculate exchange key (g^r mod p);
k_s ← calculate password key for encyrption (y^s mod p);
k_r ← calculate password key for decryption (x^r mod p);
Store k_r on wallet;

/* Calculation of salt                                           */
h ← calculate hash value for camera ID;
salt ← store digest of h ;

/* Generate encryption key and apply Fernet encryption           */
kdf ← define key derivative function PBKDF2HMAC with calculated salt;
converted_k_s ← convert k_s into byte array;
EK ← base64.urlsafe_b64encode(kdf.derive(converted_k_s));
Input: summarized video;
IV ← generate randomly;
encrypted_video ← apply Fernet encryption on summarized video by using IV and EK;
Delete EK;
Output: fully encrypted video or token;

/* Generate decryption key and apply Fernet decryption           */
Read: camera ID, k_r;
kdf ← define key derivative function PBKDF2HMAC with calculated salt;
converted_k_r ← convert k_r into byte array;
DK ← base64.urlsafe_b64encode(kdf.derive(converted_k_r));
Input: fully encrypted video or token;
decrypted_video ← apply Fernet decryption on fully encrypted video by using IV and
  DK ;
Output: summarized video;
```

The Fernet generated token was based on the concatenation of version (8 bits), timestamp (64 bits), IV (128 bits), cipher (multiple of 128 bits), and HMAC (hash-based message authentication code). The version number tells the used version of Fernet, a timestamp is the token creation time, IV is the initialization vector that was used for AES encryption, the ciphertext is the encrypted data, and authentication in Fernet was achieved through HMAC that uses *SHA256* hash function. The token generated in the encryption process was further used at the time of decryption. EK was deleted immediately after the encryption process and DK was generated in the system of an authorised stakeholder who has access to decrypt the video.

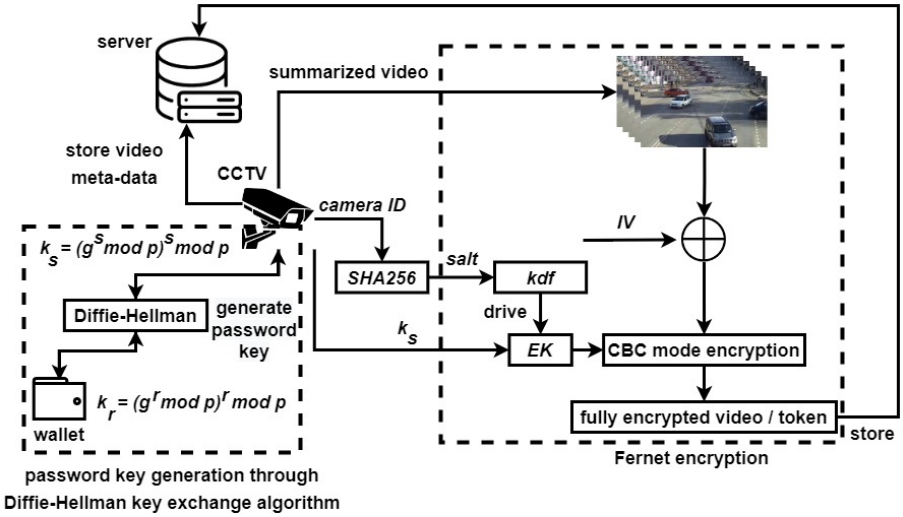


Figure 8. Integration of Diffie–Hellman key exchange algorithm with Fernet encryption.

The process shown in Figure 9 was used to retrieve original summarized video from the encrypted video by XORing the encrypted video with IV followed by decryption with DK. The summarized video was decrypted with the same key (symmetric encryption) that was used to encrypt the video. Therefore, the key (k_r) calculated through DH and stored in wallet was used to calculate decryption key (DK). Below are the details of the decryption process:

1. Authorised stakeholder accessed the fully encrypted video along with meta-data stored on the server.
2. Calculate *salt* using *camera ID*.
3. DH generated key k_r was read from the wallet and the same key generation process was repeated to generate DK.
4. IV was read from Fernet token to decrypt the video with IV and DK.
5. The summarized video was decrypted for the authorised stakeholder.

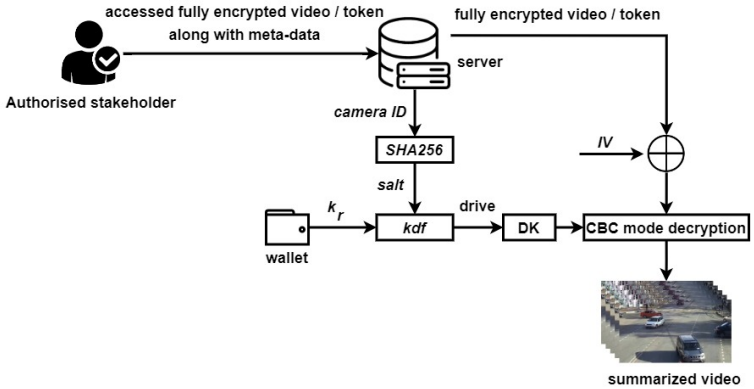


Figure 9. Decryption key generation process along with Fernet decryption.

4. Results and Discussion

For testing the trained YOLO over real-time videos footage, the system specifications were: NVIDIA Geforce RTX 3070 GPU, an Intel(R) Core(TM) i7-10700F CPU @ 2.90GHz 2.90 GHz processor, 16 GB of RAM, a 64-bit operating system, an x64-based processor, and Windows 11.

According to the details in Table 3, 0.4 ms were consumed for pre-processing, 9.3 ms for inference, and 1.2 ms for post-processing by each frame, which means on average it

took 10.9 ms to classify a frame in the video. For example, a video with 200 frames, 30 fps and 7 s length will take 2.18 s for event detection as calculated by:

$$T = F \times A \tag{4}$$

where T is the total time taken by a video to detect events, F is the total number of frames in a video footage, and A is the average time taken by each video frame to detect an event.

The accuracy of 5 videos was calculated based on bounding boxes (BB) drawn for each detected accident, as shown in Table 4. Videos were summarized only if an accident occurs, therefore, only the accident class was used for calculating accuracy of the model. The model's accuracy varies from 55% to 85%, when tested on real-time videos footage. These promising results ensure that YOLO can be trained for event detection as well. Equation (5) is used to calculate accuracy for each video, where *TrueBB* is the number of true bounding boxes detected for accidents in a video and *TotalBB* is the total number of bounding boxes detected for accidents in a video.

$$Accuracy = \frac{TrueBB}{TotalBB} \times 100 \tag{5}$$

Table 4. Test result accuracy for detected accidents in real-time test videos footage.

Video No.	Frames	BB		Total BB	Accuracy (%)
		True	False		
1	320	338	282	620	55
2	238	89	29	118	75
3	480	146	95	241	61
4	480	252	44	296	85
5	360	168	32	200	84

The number of frames of original and summarized videos are given in Table 5 along with the sizes of original, summarized and encrypted summarized videos. The results show that number of frames and sizes of summarized videos are mostly less than the original videos even if encryption is applied before storage.

Table 5. Comparison of frames and sizes in original and summarized videos footage.

Video No.	Frames		Video Size (MB)		
	Original	Summarized	Original	Summarized	Encrypted Summarized
1	320	260	9.02	6.88	9.17
2	238	118	2.99	1.45	1.94
3	480	240	9.37	4.67	6.23
4	480	270	19.9	10.8	14.4
5	360	180	6.87	3.46	4.61

Table 6 shows a reduction in the duration (video length) of every original video. According to the details, video 1 is reduced by 20%, video 2 by 51.14%, video 3 and 5 by 50%, and video 4 by 43.75%. On average, videos are reduced by 42.97%, which is quite a good reduction in the duration of any video.

Table 6. Reduction in duration of real-time test videos footage.

Video No.	Length (sec)		Reduction %
	Original	Summarized	
1	10	8	20
2	7	3	51.14
3	16	8	50
4	16	9	43.75
5	12	6	50

Key generation for encryption and decryption processes took 1.25 s on average. Table 7 shows the time (sec) consumed by encryption and decryption processes. All the videos consumed approximately 1 s for encryption and 1 s for decryption, excluding video 4, which took 1.25 s for each encryption and decryption process.

Table 7. Time (sec) consumed in the process of encryption and decryption.

Video No.	Encryption	Decryption
1	1.07	1.07
2	1.02	1.01
3	1.06	1.05
4	1.25	1.25
5	1.05	1.04

According to the test results, YOLO detects an event and draws a bounding box around the event which increases the size of video. A graph for storage space analysis of videos is shown in Figure 10. The comparison of videos in their original form is done with the videos after event detection, video summarization, and encryption. Results show that size of the original video is increased after event detection due to bounding boxes information in the videos. Furthermore, when this video is encrypted, the size is increased again. Whereas, when a summary is detected based on accidents detected in each video, the size of video is reduced. Consequently, if the encryption is applied on this summarized video, the size of video is increased but it is less than the size of the original video. The duration of real-time test videos footage is only a few seconds, therefore, the size difference is not as much notable. However, in real-time scenarios where CCTV cameras are working all day (24/7), this technique would save a large number of storage space and will also reduce the time to access the video after an event has occurred.

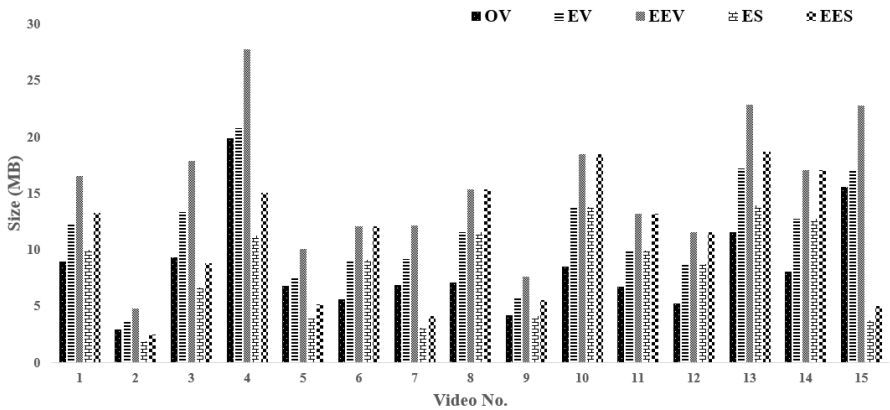


Figure 10. Size comparison through storage space utilisation (OV- original video, EV- events video, EEV- encrypted events video, ES- event based summarization, and EES- encrypted event based summarization).

5. Comparative Analysis

Table 8 shows a comparative analysis of the proposed solution with some existing techniques for video summarization using YOLO. The use-cases of other techniques for video summarization include sports, and query-based-object, and object detection, whereas our solution is focused on event detection through YOLO. Additionally, all techniques in the past have targeted video summarization, not privacy preservation. Therefore, event-based video summarization (detection of road traffic accidents) followed by encryption before storage on server is a proposed solution for smart EU-GDPR compliant systems.

Table 8. Comparison of techniques for video summarization using YOLO.

Reference	Use-Case	Event Detection	Privacy Preservation
[18]	Sports	No	No
[19]	Query based object detection	No	No
[20]	Object detection	No	No
[21]	Object detection	No	No
Proposed Framework	Road accident detection	Yes	Yes

6. Limitations and Future Work

There are certain identified limitations of the proposed solution, which are as follows:

- Unavailability of synthetic data for training on multiple road events rather than just accidents/non-accidents.
- Model is tested on fixed position CCTV recorded videos footage only.
- Only vehicular accidents are focused in this research, while pedestrian crashes are not considered.

In the extension of this research work, the security of keys can be further enhanced by integrating a standard key management protocol. Another future direction could be to take test videos from dynamic cameras, i.e., dashcams and drones to validate the trained model. A new pedestrian dataset can also be added to the existing model for training and testing to provide a solution for pedestrian accidents, which would be a promising application of the proposed work for smart vehicles.

7. Conclusions

Video surveillance is a pervasive phenomenon throughout the world and their intelligent utilisation can assist in enhancing the smartness of smart cities in terms to reduce action response time. A smart surveillance application offers many benefits, but protecting an individual's privacy along with their associated objects is equally important. In the context of regulation, data protection by design is still at an immature stage and requires reliable secure technologies for the privacy protection of digitised visual data.

To facilitate EU-GDPR compliance in smart cities' infrastructure, this paper proposed a privacy preserved solution for smart cities by training YOLO on synthetic and annotated road traffic accident data containing accident and non-accident events for video summarization. The trained YOLO model is tested on real-time CCTV videos footage, but to protect the sensitive event records, symmetric encryption with Fernet cipher is applied on the summarized videos along with Diffie–Hellman key exchange algorithm, and *SHA256* hash algorithm. The encryption key is deleted immediately after the encryption process and the key stored in a hardware wallet is later on used by the authorised stakeholders to generate a decryption key to avoid MITM attacks on keys. The results calculated on five (5) test videos show that the model achieved 55% to 85% accuracy for event (accident) detection and this accuracy is measured by the number of true bounding boxes drawn on each accident area. The accuracy of the model can be increased by increasing the training dataset, as this research is conducted on limited synthetic training data available for accidents and non-accidents road traffic events.

Author Contributions: Conceptualization, M.T., M.N.A. and N.K.; methodology, M.T. and M.N.A.; software, M.T.; validation, M.T.; formal analysis, M.T. and M.N.A.; investigation, M.T.; resources, N.K. and B.L.; data curation, M.T.; writing—original draft preparation, M.T. and M.N.A.; writing—review and editing, N.K., Y.Q. and B.L.; visualization, M.T.; supervision, M.N.A. and Y.Q. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The synthetic dataset for the training of YOLO model was taken from a public source, BeamNG drive [29]. After annotation of synthetic dataset, it is made publicly available at https://www.kaggle.com/datasets/mehwishtahir722/accident-and-nonaccident-dataset-for-yolo (accessed on 11 November 2022).

Acknowledgments: This work is part of doctoral degree research funded by the Presidential Doctoral Scheme (PDS) of the Technological University of the Shannon: Midlands Midwest (Athlone Campus), N37 HD68 Athlone, Ireland.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Shifa, A.; Asghar, M.N.; Noor, S.; Gohar, N.; Fleury, M. Lightweight Cipher for H. 264 Videos in the Internet of Multimedia Things with Encryption Space Ratio Diagnostics. *Sensors* **2019**, *19*, 1228. [CrossRef]
2. Wang, Z.; Liu, J. A Review of Object Detection Based on Convolutional Neural Network. In Proceedings of the 2017 36th Chinese Control Conference (CCC), Dalian, China, 26–28 July 2017; pp. 11104–11109.
3. Chen, C.; Liu, M.-Y.; Tuzel, O.; Xiao, J. R-CNN for Small Object Detection. In Proceedings of the Asian Conference on Computer Vision, Taipei, Taiwan, 20–24 November 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 214–230.
4. Girshick, R. Fast R-Cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
5. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-Cnn: Towards Real-Time Object Detection with Region Proposal Networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*. [CrossRef]
6. Zaidi, S.S.A.; Ansari, M.S.; Aslam, A.; Kanwal, N.; Asghar, M.; Lee, B. A survey of modern deep learning based object detection models. *Digit. Signal Process.* **2022**, *126*, 103514.
7. Liu, C.; Tao, Y.; Liang, J.; Li, K.; Chen, Y. Object Detection Based on YOLO Network. In Proceedings of the 2018 IEEE 4th Information Technology and Mechatronics Engineering Conference (ITOEC), Chongqing, China, 14–16 December 2018; pp. 799–803.
8. Aslam, A.; Curry, E. Towards a generalized approach for deep neural network based event processing for the internet of multimedia things. *IEEE Access* **2018**, *6*, 25573–25587.
9. Santad, T.; Silapasupphakornwong, P.; Choensawat, W.; Sookhanaphibarn, K. Application of YOLO Deep Learning Model for Real Time Abandoned Baggage Detection. In Proceedings of the 2018 IEEE 7th Global Conference on Consumer Electronics (GCCE), Nara, Japan, 9–12 October 2018; pp. 157–158.
10. Liu, R.; Ren, Z. Application of Yolo on Mask Detection Task. In Proceedings of the 2021 IEEE 13th International Conference on Computer Research and Development (ICCRD), Beijing, China, 5–7 January 2021; pp. 130–136.
11. Laroca, R.; Severo, E.; Zanlorensi, L.A.; Oliveira, L.S.; Gonçalves, G.R.; Schwartz, W.R.; Menotti, D. A Robust Real-Time Automatic License Plate Recognition Based on the YOLO Detector. In Proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil, 8–13 July 2018; pp. 1–10.
12. Cao, Z.; Liao, T.; Song, W.; Chen, Z.; Li, C. Detecting the Shuttlecock for a Badminton Robot: A YOLO Based Approach. *Expert Syst. Appl.* **2021**, *164*, 113833.
13. Park, S.-S.; Tran, V.-T.; Lee, D.-E. Application of Various Yolo Models for Computer Vision-Based Real-Time Pothole Detection. *Appl. Sci.* **2021**, *11*, 11229. [CrossRef]
14. Asghar, M.N.; Kanwal, N.; Lee, B.; Fleury, M.; Herbst, M.; Qiao, Y. Visual Surveillance within the EU General Data Protection Regulation: A Technology Perspective. *IEEE Access* **2019**, *7*, 111709–111726. [CrossRef]
15. Asghar, M.N.; Ansari, M.S.; Kanwal, N.; Lee, B.; Herbst, M.; Qiao, Y. Deep Learning Based Effective Identification of EU-GDPR Compliant Privacy Safeguards in Surveillance Videos. In Proceedings of the 2021 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech), In Virtual, 25–28 October 2021; pp. 819–824.
16. Tahir, M.; Asghar, M.N.; Kanwal, N.; Lee, B.; Qiao, Y. Joint Crypto-Blockchain Scheme for Trust-Enabled CCTV Videos Sharing. In Proceedings of the 2021 IEEE International Conference on Blockchain (Blockchain), Melbourne, Australia, 6–8 December 2021; pp. 1–6.
17. Simmons, G.J. Symmetric and Asymmetric Encryption. *ACM Comput. Surv. CSUR* **1979**, *11*, 305–330. [CrossRef]
18. Guntuboina, C.; Porwal, A.; Jain, P.; Shingrakhia, H. Deep Learning Based Automated Sports Video Summarization Using YOLO. *ELCVIA Electron. Lett. Comput. Vis. Image Anal.* **2021**, *20*, 99–116.

19. Kakodra, S.S.; Sujatha, C.; Desai, P. Query-By-Object Based Video Synopsis. In Proceedings of the 2021 International Conference on Intelligent Technologies (CONIT), Karnataka, India, 25–27 June 2021; pp. 1–5.
20. Ul Haq, H.B.; Asif, M.; Ahmad, M.B.; Ashraf, R.; Mahmood, T. An Effective Video Summarization Framework Based on the Object of Interest Using Deep Learning. *Math. Probl. Eng.* **2022**, *2022*, 7453744. [CrossRef]
21. Negi, A.; Kumar, K.; Saini, P.; Kashid, S. Object Detection based Approach for an Efficient Video Summarization with System Statistics over Cloud. In Proceedings of the 2022 IEEE 9th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON), Allahabad, India, 2–4 December 2022; pp. 1–6.
22. Abu Taha, M.; Hamidouche, W.; Sidaty, N.; Viitanen, M.; Vanne, J.; El Assad, S.; Déforges, O. Privacy Protection in Real Time HEVC Standard Using Chaotic System. *Cryptography* **2020**, *4*, 18. [CrossRef]
23. Shifa, A.; Imtiaz, M.B.; Asghar, M.N.; Fleury, M. Skin Detection and Lightweight Encryption for Privacy Protection in Real-Time Surveillance Applications. *Image Vis. Comput.* **2020**, *94*, 103859.
24. Alawi, A.R.; Hassan, N.F. A Proposal Video Encryption Using Light Stream Algorithm. *Eng. Technol. J.* **2021**, *39*, 184–196. [CrossRef]
25. Huang, X.; Arnold, D.; Fang, T.; Saniie, J. A Chaotic-Based Encryption/Decryption System for Secure Video Transmission. In Proceedings of the 2021 IEEE International Conference on Electro Information Technology (EIT), Mt. Pleasant, MI, USA, 14–15 May 2021; pp 369–373.
26. Tyagi, S.S. Enhancing Security of Cloud Data through Encryption with AES and Fernet Algorithm through Convolutional-Neural-Networks (CNN). *Int. J. Comput. Netw. Appl.* **2021**, *8*, 288–299.
27. Li, H.; Xiezhang, T.; Yang, C.; Deng, L.; Yi, P. Secure Video Surveillance Framework in Smart City. *Sensors* **2021**, *21*, 4419. [CrossRef]
28. Aribilola, I.; Asghar, M.N.; Kanwal, N.; Fleury, M.; Lee, B. SecureCam: Selective Detection and Encryption enabled Application for Dynamic Camera Surveillance Videos. *IEEE Trans. Consum. Electron.* **2022**. [CrossRef]
29. Home. Available online: https://www.beamng.com/game/ (accessed on 29 May 2022).
30. Make Sense. Available online: https://www.makesense.ai/ (accessed on 30 June 2022).
31. Accident and Non-Accident Dataset for YOLO. Available online: https://www.kaggle.com/datasets/mehwishtahir722/accident-and-nonaccident-dataset-for-yolo (accessed on 11 November 2022).
32. Ultralytics/YOLO*v*5. Ultralytics. Available online: https://github.com/ultralytics/YOLOv5/blob/c98128fe71a8676037a0605ab389c7473c743d07/README.md (accessed on 4 October 2022).
33. Google Colaboratory. Available online: https://colab.research.google.com/ (accessed on 4 June 2022).
34. Car Crashes Time, CCTV CAR CRASHES COMPILATION 2018 #EP. 20, 8 January 2018. Available online: https://www.youtube.com/watch?v=gQkoujWBxqg&t=452s (accessed on 23 August 2022).
35. Fernet (Symmetric Encryption) — Cryptography 39.0.0.dev1 Documentation. Available online: https://cryptography.io/en/latest/fernet/ (accessed on 18 November 2022).
36. Mathews, S.P.; Gondkar, R.R. Protocol Recommendation for Message Encryption in Mqtt. In Proceedings of the 2019 International Conference on Data Science and Communication (IconDSC), Bangalore, India, 1–2 March 2019; pp. 1–5.
37. Rescorla, E. Diffie-Hellman Key Agreement Method. 1999. Available online: https://www.rfc-editor.org/rfc/rfc2631.html (accessed on 18 November 2022).
38. Lehto, N.; Halunen, K.; Latvala, O.-M.; Karinsalo, A.; Salonen, J. CryptoVault-A Secure Hardware Wallet for Decentralized Key Management. In Proceedings of the 2021 IEEE International Conference on Omni-Layer Intelligent Systems (COINS), Barcelona, Spain, 23–25 August 2021; pp. 1–4.
39. Asghar, M.N.; Ghanbari, M. MIKEY for keys management of H. 264 scalable video coded layers. *J. King Saud-Univ.-Comput. Inf. Sci.* **2012**, *24*, 107–116.