Team 7 - Privacy-Preserving Sharing: Empowering Users While Safeguarding Privacy

Kevin Chen, John Daniel, Claire Kim, Kyle Smith

CMSC 491/691 Data Privacy

May 2024

Abstract

In today's digital landscape, the enormous amount of data presents both opportunities and challenges, particularly in safeguarding privacy while harnessing the benefits of data sharing for positive use cases. This paper delves into the development and proposal of a privacy-preserving sharing framework tailored for video data from doorbell cameras, aiming to empower users to share such data with interested parties while maintaining comprehensive privacy safeguards. Building upon existing technologies such as motion detection and other anomaly detection techniques, we propose a novel framework integrating machine learning algorithms for enhanced post-processing capabilities. This framework enables interested individuals to query a network of doorbell cameras for shared doorbell videos, retrieving privacy-protecting versions that aligns with the parameters set by the video owner. We examine potential applications, acknowledge limitations, and discuss avenues for future enhancements within the scope of this framework.

Contents

1 Introduction

Recent years have shown an increase in video doorbell cameras in U.S. households, with about 20% of the population having them installed. One of the largest video doorbell cameras currently on the market is Ring, which is owned by Amazon.

With over one million vehicles being stolen the past year in 2023, averaging at about 2052 a day, and 85 cars an hour, law enforcement has began resulting to home doorbell cameras for footage. When community members report that their vehicle was stolen, law enforcement requests neighbors for possible footage, with vague information. The video doorbell owners are given a time frame, and possibly a car type and color to be looking out for. Parsing the several hours of video footage to look for specific frames is time consuming, and could result in no frames being found, or sensitive personal information within the found frames.

We want to introduce a framework that will allow users to parse through their video footage and look for specific car frames and reduce the it to only consist of the relevant frames. The framework will benefit the users by ensuring that unwanted frames do not accidentally get selected due to one rushing through the long footage time frames. The parsing will be done in a quick manner, and allow the user to individually view each selected clip before sharing the video evidence.

2 Related Work

Some prior research has been done with the Heidelberg Collaboratory where they parsed through video frames to look for any abnormal activity, such as running, a car where only people walk (ex. sidewalk), individuals walking on grass. This is a starting point for our framework as it addresses and chooses frames for activities that may raise suspicion and help filter them out. For our framework, we do not only want to look for alarming frames, but want to be able to request a query to look for specific frames such as ”a tall male with a blue shirt” or ”white car crashing into mailbox”.

Other areas that have been researched briefly in the past are parsing videos for cars to get the license plate, this is a feature that we plan to implement within our framework, but we were able to incorporate similar models to detect car movements and frames within footage to identify the necessary captures.

3 Problem & Contribution

The problem we address revolves around the challenge of efficiently sharing valuable doorbell camera video data while preserving owner privacy. Consider scenarios where individuals possess video footage, from doorbell cameras like Amazon Ring, Google Nest, or Arlo Essentials, which could be relevant to interested individual’s objectives, such as law enforcement investigations. However, existing methods lack sufficient privacy measures, leading to potential risks. For instance, when approached by authorities seeking footage of a specific event, users face dilemmas: sharing the footage may divulge unnecessary personal information, jeopardizing privacy and safety. Moreover, repeated requests from law enforcement can become burdensome.

We propose a framework where an interested party can send a query to a network of opt-in doorbell cameras that have the ability to run a machine learning algorithm and perform post video processing before sending the data to the interested party. Our framework aims to mitigate some concerns presented in the paragraph above by facilitating automated secure video sharing through a dedicated platform that allows for the data owner to customize who has access to improve the level of privacy. By employing advanced video and metadata modification techniques, our framework aims to retain only pertinent content while obfuscating sensitive details. This approach ensures that shared footage maintains its utility for external parties while safeguarding user privacy.

The steps of the proposed framework are briefly as follows:

1. Video owners opt-in to the doorbell camera sharing network and define their sharing preferences as outlined in the Privacy Enhancing Techniques & Implementation {4} section on Access Control.

2. Interested parties are validated and register with specified roles and submit queries to the system.

3. Queries from interested parties are relayed to a network of opted-in doorbell cameras, which accept or deny requests based on Access Controls set by the video owner according to the interested party's role.

4. Each opted-in doorbell camera autonomously utilize a pretrained machine learning algorithm on its collected motion or anomaly-detected footage to determine query matches. This approach mirrors the benefits of Federated Learning, eliminating the need to transmit preprocessed data to anyone, including the interested parties.

5. In the absence of matching footage, a response is sent to the interested party indicating no results were found. If matching footage is identified, initial metadata modification is performed as described in the Privacy Enhancing Techniques & Implementation {4} section on De-Identification.

6. Further De-Identification is conducted using another pretrained machine learning algorithm to remove and obfuscate sensitive details unrelated to the requested query such as person, items, and more.

7. Finally, the modified video is transmitted back to the interested party.

Note: All communications and data shall be encrypted at rest and/or in motion, in accordance with the Privacy Enhancing Techniques & Implementation {4} section on Encryption.

4 Privacy Enhancing Techniques & Implementation

Given the constraints of time and our team's limited experience in implementing machine learning techniques, we regrettably could not implement and experiment with all the proposed privacy-enhancing techniques and implementation outlined in the following sections. It is imperative to acknowledge that further research and development are warranted that incorporates all the essential privacy safeguards into our proposed framework. In the final section titled "Implementation," we discuss the conceptual implementation of the framework utilizing all the proposed Privacy Enhancing Techniques (PETs), underscoring our commitment to addressing these vital privacy considerations in future iterations of the project

4.1 Encryption

Encryption serves as a cornerstone of data privacy, as it provides the necessary means to enforce privacy measures. By employing encryption, we ensure that users' data remains inaccessible to unauthorized individuals and grant users control over data access. For instance, in our proposed framework, securing the video data is critical to ensure that only authorized parties, such as the video's owner or designated recipients, possess the decryption key to be able to view and use the data.

Effective encryption encompasses multiple stages, including encryption at rest and encryption in motion. In our purposed framework, we prioritize both encryption at rest and encryption in motion. Locally encrypting videos ensures their security while being stored, rendering them unusable to unauthorized entities attempting to obtain copies without authorization. Additionally, during data transmission, we employ encryption protocols to safeguard against interception, ensuring that only the intended recipient, who queried the data, receives access upon decryption. We also will be encrypting the queries that are being sent out to the network, so privacy is maintained for what information is being asked by the interested party.

This comprehensive encryption strategy fortifies data privacy throughout its lifecycle, from storage to transmission, thereby upholding the confidentiality and integrity of owners' valuable video data, as well as the interested partys' queries.

4.2 Access Control

Access control plays a pivotal role in our proposed framework, granting our system the capability to regulate resource access. While various methods exist for implementing access control, our framework advocates for the adoption of Role-Based Access Control (RBAC) alongside Discretionary Access Control (DAC). RBAC with DAC will involve empowering resource owners to grant access and permissions based on the roles of interested parties. In our context, we propose the establishment of distinct roles for different stakeholders, such as authorities and media entities, then enabling video data owners to exert precise control over data access based on those roles.

By leveraging RBAC with DAC, our framework not only strengthens privacy measures but also empowers video data owners with granular control. Owners can delineate access privileges according to specific roles, ensuring that only permitted parties of their choice can access their data. This approach provides a robust privacy measure where access is tailored to the unique needs and permissions of each video owner, enhancing data security and confidentiality for the video owners.

A potential alternative for implementing access control, especially if you seek more granular control over access permissions, involves considering the use of Attribute-Based Access Control (ABAC) alongside Discretionary Access Control (DAC). ABAC is a method of access control that determines authorization for operations by evaluating attributes associated with the subject, object, requested operations, and environmental factors. In our context, ABAC would afford video owners greater precision in controlling access to their data and specifying the types of queries that can be made. This approach offers a flexible and dynamic means of managing access rights, enabling video owners to tailor access permissions based on specific attributes, enhancing overall security and privacy measures.

4.3 De-Identification

De-identification is paramount for bolstering data privacy by limiting the amount and type of data that is published. Through techniques like modification, masking, and data removal, we aim to mitigate the risk of identifying specific data points of user video footage. Our approach primarily targets the cleanup of metadata associated with the footage, eliminating device-specific details such as IP addresses, MAC addresses, and camera types, alongside personal identifiers like the owner's name and age. By anonymizing this information, we ensure that the footage remains devoid of any identifiable markers, thus reinforcing privacy safeguards. Additionally, for scenarios involving multiple video data points, de-identification serves to obscure the origin of the footage, further enhancing anonymity/privacy. We also propose masking and modification strategies to obscure sensitive details, such as substituting specific addresses with generic street information. While these techniques may initially appear to pose a trade-off between usability and privacy, they provide an additional layer of anonymity, significantly strengthening privacy protection.

In our proposed framework, we leverage machine learning algorithms to automate the de-identification process further. We will have machine learning algorithms that are adept at selectively blurring out non-essential information from the video data, such as faces or other identifiable features, based on the query made by the interested party. For instance, if the query pertains to identifying cars within a specific timeframe, the algorithm intelligently blurs irrelevant details while retaining pertinent information about the vehicles. We also seek to enhance the de-identification approach by creating or utilizing a machine learning algorithm to recognize common individuals appearing in the video footage. This adaptive capability ensures continuous privacy protection by automatically blurring out familiar faces during requested queries, unless explicitly permitted by the owner.

By striking a balance between privacy preservation and data utility, our approach maximizes the efficacy of shared data while upholding stringent privacy standards.

4.4 Implementation

The ideal implementation of our framework operates as follows: Video owners who opt-in understand and accept the trade-off between providing data and privacy reduction, knowing that we have implemented methods like access control, de-identification, and encryption to mitigate privacy risks while preserving utility. Upon opting into the network of doorbell cameras, owners are presented with the option of choosing between Role-Based Access Control (RBAC) or Attribute-Based Access Control (ABAC) to manage permissions for interested users. With RBAC, owners define roles that can request information from their camera, while ABAC allows for more granular control over queries and access based on specific attributes. Once access control preferences are set, the system awaits TLS 1.3 encrypted queries for further processing.

Upon receiving encrypted queries, the system searches through encrypted video data using a pre-trained machine learning agent to identify matching footage. If footage is found, the system performs three functions: first, it cleans up metadata that could potentially reveal the source of the data, such as IP and MAC addresses, and modifies location metadata to obscure precise addresses. Second, another machine learning agent processes the matching video data, blurring or obfuscating any irrelevant portions based on the submitted queries. Third, a locally trained machine learning agent identifies common items or persons in the video and blurs or obfuscates those areas as well.

Once de-identification is complete, the data is sent to authorized interested users via TLS 1.3 encryption, ensuring secure transmission. This comprehensive approach to privacy protection and data utility ensures that video owners can confidently share footage while minimizing privacy risks and preserving the integrity of their data.

Due to time constraints, we prioritized the implementation of the core features of de-identification. Details regarding how we implemented these features and the results we collected can be found in the following section below.

5 Experimentation & Methodology

5.1 Technology Used

The pipeline was developed using Python and leveraged the Open Computer Vision Library (OpenCV) to operate on frames from a video. Two computer vision models were employed: one for detecting the presence of cars and another for matching images containing cars to a specified input prompt. The haarcascade_cars.xml model from Car-Detection-OpenCV and the Fast Segment Anything Model (FastSAM) in text prompt mode were utilized for processing frames. If both models selected a frame, it was reassembled into an output video using OpenCV functions and saved to be later sent to the authorized party.

The parameters in haarcascade_cars.xml were specifically trained to identify cars from multiple angles, operating at near real-time on 30 fps 1080p video. However, it exhibited a higher rate of false positives, occasionally identifying cars where there were none. FastSAM, a general-purpose segmentation model, includes an option for using a text prompt to segment an image. This feature enabled users to search for specific, non-predetermined objects in the recording by inputting text arguments into the application. While effective, FastSAM's text prompt mode has a substantially higher processing cost. On an Intel i9-12900K CPU, processing a single frame could take up to one second. By reducing the total number of frames processed by FastSAM, the application's performance was significantly enhanced.

5.2 Data Pipeline

After data is recorded by a camera, it can be stored locally on the camera or in a centralized storage system managed by the building or the camera user. The storage system awaits queries that specify the timestamp interval of the desired footage and the objects to search for in the frames. Using the provided timestamp, the system identifies the appropriate file and processes frames until the end of the footage or the specified timestamp interval is reached.

Frames are analyzed by the computer vision models, which either reject or accept them. Accepted frames are compiled into an output video. The output video includes no metadata except for the timestamp and is saved on the system, enabling the user to share it with approved parties.

5.3 Processing Pipeline

The processing pipeline comprises four steps, with each frame processed individually, independent of information from other frames. Once initiated, the process identifies the footage file based on the provided timestamp arguments and loads the file for processing. The first frame at the specified timestamp undergoes preprocessing, which includes downscaling and converting the frame to grayscale for model analysis.

The first model, using the haarcascade_cars parameters, analyzes the grayscale copy of the frame. If a car is detected, the original color frame is passed to the FastSAM model along with a copy of the input prompt. If the FastSAM model segments an area of the frame that matches the text prompt, the frame is saved into the output video.

The pipeline operates exclusively on individual frames, preserving only the necessary timestamps from the original video. No metadata or sound from the original video is retained or added to the output video, ensuring that the only information included is the time the video was recorded, helping to increase privacy.

6 Results

The dataset was designed to test the accuracy of the model in detecting cars and specific attributes of cars, such as color. It consisted of eight different video segments categorized by the following types: red cars in the video, cars in the video but not red, videos shot at night/day, and videos with no cars. Each video was tested with two prompts: "cars" and "red cars." These prompts instructed the application to find all frames with cars or specifically frames where red cars were present, enabling an assessment of the system's performance and accuracy with more complex queries.

In total, 123,994 frames were processed, resulting in a 45% reduction to 56,217 frames after post-processing. The overall true positive rate was 96.66%, and the true negative rate was 96.23%. The false positive rate was 4.79%, and the false negative rate was 3.51%.

For the "cars" prompt, the frames were reduced from 62,448 to 45,853, a reduction of 26.65%. This prompt yielded a true positive rate of 96.37% and a true negative rate of 96.51%, with false positive and false negative rates of 1.31% and 3.77%, respectively.

The "red cars" prompt aimed to return all frames where a red car was present. This prompt resulted in a frame reduction from 62,448 to 10,364, representing an 83.40% reduction. The true positive rate for this prompt was 97.69%, and the true negative rate was 96.14%. The false positive rate was 3.35%, and the false negative rate was 2.37%. The increased specificity of the "red cars" prompt significantly improved frame reduction compared to the "cars" prompt, enhancing the model's privacy performance.

To evaluate the framework's performance with realistic footage, doorbell and security camera videos posted online were collected. These videos included side profiles of vehicles typical for suburban homes, pedestrians, various non-car objects, and both day and night footage. Some samples included news broadcasts of crimes captured by security cameras or doorbell cameras, demonstrating the model's ability to exclude frames without cars.

Seventeen tests were conducted on footage ranging from 20 seconds to 30 minutes, using the prompts "cars" and "red cars." After processing, the footage was analyzed to determine the number of frames that matched the prompt criteria.

The privacy utility of the model, indicated by frame reduction and true positive percentages, showed a significant improvement with the "red cars" prompt, reducing frames by 83.40% compared to the 26.65% reduction with the "cars" prompt. Despite the increased frame reduction, there were no significant changes in accuracy performance between the two prompts, demonstrating the model's robustness in maintaining high accuracy while improving privacy through specific prompts.

6.1 Privacy Concerns

Privacy concerns are paramount in our application. We do not collect user information, nor do we process or track how users interact with the application. There is no processing or storage of any user information externally. All data processed on the user's machine remains solely on that machine, ensuring user privacy. The libraries utilized in this project, including OpenCV and FastSAM, are selected with privacy in mind. OpenCV, a crucial component of our project, is an open-source library designed to process all information locally, ensuring that user data remains within their control. Similarly, FastSAM operates entirely on the user's machine, further safeguarding user privacy. Our commitment to privacy extends throughout the application, prioritizing user confidentiality and data security at every stage.

7 Conclusion

In this paper, we propose a framework where an interested party, such as law enforcement can send a query to a server with video doorbell cameras to receive footage of possible evidence. The proposed framework takes footage of various lengths and processes the frames to conclude of only the requested frames. The tested framework has an average of 96.609% accuracy rate at detecting "cars" and "red cars". Although we mainly focused on cars within the proposed framework, we plan to incorporate detecting people and other elements within our future work.

7.1 Future Work

Due to semester time constraints some of the proposed features of the frameworks were unimplemented, but in the future some of the main things we would improve are the implementation of more PET's, for example, De-Identification, to maintain one's privacy preferences. Some other feature's that would be implemented are the ability to search for more specific details other than "car" and "red car", and for frames that may consist "boy with green jacket" or "motorcycle".

Currently, the pipeline takes about 7.2 hours to process a 30 minute video, which means it takes about 14.2 minutes for the current implemented system to analyze 1 minute worth of footage. Although the model was able to identify the clips with about 96.609% accuracy rate, 7.2 hours is quite a long time to wait if the possible evidence is urgently requested. In the improved framework, we plan to improve the current algorithm and streamline process to make it easily accessible for the users.

Once we have the improved system, we will retest the model accuracy with various clips that were not collected solely from Ring news footage, or footage of cars through neighborhoods as some of them may have skewed some of the accuracy rates.

8 Contributions

8.1 Kevin Chen

As project manager, Kevin organized sprint planning and managed tasks to optimize productivity. Setting up the coding repository and sprint board facilitated seamless code development and progress tracking. Kevin's initial research into computer vision models helped decide the project's direction. His research into Privacy Enhancing Techniques (PETs) provided information and insights for our project's final direction and proposed framework. Kevin refined project goals and objectives, designed the proposed framework, and developed project ideas. Kevin also provided coordination and leadership on all deliverables, ensuring project cohesion and timeliness. He aimed to maintain a balanced project scope and goals. Kevin helped outline the presentation and provided finalized contents for slides Approach, Contributions, and Privacy Enhancing Technologies. Kevin outlined the final paper, ensuring clarity and coherence in presenting our research and proposed framework. Additionally, he authored and finalized key sections of the paper, including the Abstract, Problem & Contribution, and Privacy Enhancing Techniques & Implementation. Lastly, Kevin reviewed, edited, and incorporated all peer components of the final research paper into the final document.

8.2 Claire Kim

Claire was in charge of researching doorbell cameras privacy policies and collecting video footage to feed into the model. She worked closely with Kyle to provide any necessary materials to setup the model and pipeline testing. Claire assisted with redefining the project scope and goals when directions varied throughout the semester. She was in charge of the final presentation layout and finalizing information throughout the slides. She authored the Introduction, Related work, Conclusion, Future work, and co-authored Contributions. Claire worked closely with Kevin to finalize the research paper and work through the theoretical side of the research.

8.3 Kyle Smith

Kyle was responsible for organizing the design and development of the pipeline and oversaw the technology components for the GitHub repository. He tested various available computer vision models, selected the final model parameters for frame processing, and managed their integration into the pipeline. This involved fine-tuning the model parameters to ensure the application met the project's privacy goals, as well as writing the code for processing frames from the original video through the models to the final set of frames. Kyle also conducted comprehensive testing of the completed pipeline to generate final data on its accuracy and precision. His contributed to the experimentation and methodology sections of the paper, where he detailed the processes and results of the model evaluations.

8.4 John Daniel

John conducted extensive research and testing on various AI models to determine their suitability for the project. He developed robust functionality for handling command line arguments. John was also responsible for managing timestamps and implementing the conversion from relative to absolute timestamps, ensuring accuracy and consistency in data processing. John also merged frames back into video format for output, facilitating both video and image creation to accommodate analysis and testing requirements. John helped with updating the README and creating the requirements file. John drafted the results section, privacy concerns, and future considerations for the final paper.