Journal of Electronic Imaging

JElectronicImaging.org

Video redaction: a survey and comparison of enabling technologies

Shagan Sah
Ameya Shringi
Raymond Ptucha
Aaron Burry
Robert Loce

SPIE. IS&T imaging.org

Shagan Sah, Ameya Shringi, Raymond Ptucha, Aaron Burry, Robert Loce, "Video redaction: a survey and comparison of enabling technologies," *J. Electron. Imaging* **26**(5), 051406 (2017), doi: 10.1117/1.JEI.26.5.051406.

Journal of Electronic Imaging 26(5), 051406 (Sep/Oct 2017)

REVIEW

Video redaction: a survey and comparison of enabling technologies

Shagan Sah,[a,*] Ameya Shringi,[a] Raymond Ptucha,[a] Aaron Burry,[b] and Robert Loce[c]
[a]Rochester Institute of Technology, Rochester, New York, United States
[b]Conduent Labs, Webster, New York, United States
[c]Datto Inc., Rochester, New York, United States

Abstract. With the prevalence of video recordings from smart phones, dash cams, body cams, and conventional surveillance cameras, privacy protection has become a major concern, especially in light of legislation such as the Freedom of Information Act. Video redaction is used to obfuscate sensitive and personally identifiable information. Today's typical workflow involves simple detection, tracking, and manual intervention. Automated methods rely on accurate detection mechanisms being paired with robust tracking methods across the video sequence to ensure the redaction of all sensitive information while minimizing spurious obfuscations. Recent studies have explored the use of convolution neural networks and recurrent neural networks for object detection and tracking. The present paper reviews the redaction problem and compares a few state-of-the-art detection, tracking, and obfuscation methods as they relate to redaction. The comparison introduces an evaluation metric that is specific to video redaction performance. The metric can be evaluated in a manner that allows balancing the penalty for false negatives and false positives according to the needs of particular application, thereby assisting in the selection of component methods and their associated hyperparameters such that the redacted video has fewer frames that require manual review. *© 2017 SPIE and IS&T* [DOI: 10.1117/1.JEI.26.5.051406]

Keywords: surveillance; video redaction; privacy protection; object detection; tracking.
Paper 170098SS received Feb. 9, 2017; accepted for publication Jun. 22, 2017; published online Jul. 20, 2017.

1 Introduction

A new era of surveillance has been ushered in due to advances in camera, data storage, and communications technology coupled with concerns for public safety, police-public relations, and cost effective law enforcement. According to IHS,[1] there were 245 million professionally installed video surveillance cameras active and operational globally in 2014. While the majority of these cameras were analog, over 20% were estimated to have been network cameras and around 2% were HD CCTV cameras. The number of cameras installed in the field is expected to increase by over 10% per year through 2019. Traditional police static video surveillance systems typically consist of networks of linked cameras mounted at fixed locations in public spaces such as transportation terminals, walkways, parks, and government buildings. There is also a great increase of law enforcement cameras on mobile platforms, such as dash cams and body cams. A 2013 Bureau of Justice Statistics release indicated that 68% of the 12,000 local police departments used in-car cameras.[2] A survey of large city and county law enforcement agencies on body-worn camera technology indicated that 95% planned to deploy body cameras.[3] These public, as well as private, surveillance systems have been a great aid in identifying and capturing suspects, as well as revealing behaviors between the public and law enforcement.

This vast amount of video data being collected poses a challenge to the agencies that store the data. Records created and kept in the course of government business must be disclosed under right-to-know laws unless there is an exception that prevents disclosure. The Freedom of Information Act (FOIA), 5 U.S.C. 552, is a federal law that establishes a presumption that federal governmental information is available for public review. Under FOIA, federal agencies are required to issue a determination on a disclosure within 20 working days of receipt of the request or appeal, which can be extended by 10 days in circumstances such as the request is for a significant volume of records or requires collection of records from different offices or consultation with another agency. State law enforcement agencies operate under similar disclosure guidelines.

Video recordings are considered public records for the purpose of right-to-know laws, and privacy is one exception to reject a request for disclosure. Rather than an outright rejection of a complete video record of an event, it is becoming common practice to redact portions of the video record.

Redaction protects the privacy and identity of victims, innocent bystanders, minors, and undercover police officers. Redaction involves obscuring identifying information within individual video frames. Identifying information often includes but is not limited to faces, license plates, identifying clothing, tattoos, house numbers, and computer screens. Obscuring the information typically involves blanking out or blurring the information within a region. The region could be a tight crop around a person's face, for example. Alternatively, redaction could involve blanking out the entire frame except portions that are not considered private. An example of redaction is shown in Fig. 1, where blurring is used to obfuscate the body. Figure 2 shows the high level process of releasing a redacted video.

*Address all correspondence to: Shagan Sah, E-mail: sxs4337@rit.edu

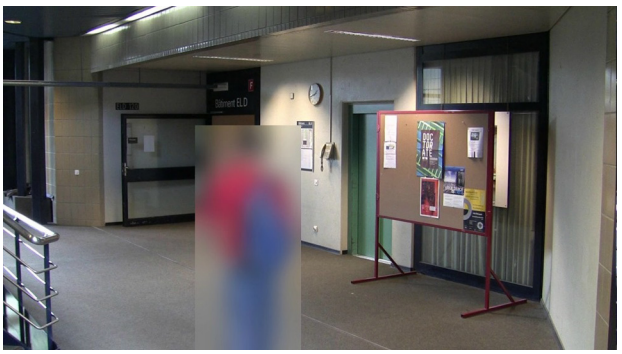


Fig. 1 Example of redaction with blurring.

The present paper reviews the video redaction problems and challenges, rigorously explores detection and tracking methods to enable redaction, and introduces a detection and tracking metric specifically relevant to redaction. The remainder of the introduction reviews current redaction practices. Section 2 reviews the two main components of a redaction system: object detection and object tracking. Section 3 presents a metric for evaluating the redaction system. Section 4 examines various types of obfuscations, and Sec. 5 discusses open problems in the redaction space.

1.1 *Current Approaches to Video Redaction*

Various approaches have been proposed and applied to privacy protection in videos. The most common ones apply visual transformations on image regions that contain the private or personally identifiable information (PII). These obfuscations can be as simple as replacing or masking faces with shapes in video frames.[4] Other common obfuscations hide objects by blurring, pixelation, or interpolation with the surroundings. More advanced techniques utilize edge and motion models for the entire video to obscure or remove the whole body contour from the video.[5] Some approaches involve scrambling the part of the image using a secret encryption key to conceal identity.[6] Korshunov and Ebrahimi[7] show the use of image warping on faces for detection and recognition. Although some studies have also shown the use of RFID tags for pinpointing the location of people in space,[8] most studies rely on image-based detection and tracking algorithms to localize an object in a video frame.

Recently, Corso et al.[9] presented a study with analysis on privacy protection in law enforcement cameras. The redaction process can be very time and labor intensive and thus expensive for a law enforcement agency. Improvements in automated redaction offer the potential to greatly relieve this cost burden. The three primary steps in a process using automation are localization of object(s) to be redacted, tracking of these objects over time, and their obfuscation. While these steps can be fully performed manually with video editing tools, current approaches are moving toward semiautomatic redaction with a manual review with extensive manual editing, which is necessary as less than perfect obfuscation of an object in even a single video frame may expose the identity and hence defeat the entire purpose of privacy protection. For example, there are commercially available tools that offer basic video redaction functionality. They have a friendly interface that gives the user the ability to manually tag the object of interest. The tagged object can then be tracked in a semiautomatic fashion through portions of the video. However, detection and tracking performance limitations still typically require a manual review to verify the final redaction.

In some existing tools, there is automated face detection, but it is limited to frontal faces and fails with occlusions, side views, size, or low-resolution images. Another common option in existing redaction tools is a color-based skin detection option; however, their efficacy with different color skins is limited. Automatic blur of the entire image is also available, but it reduces any contextual meaning in the image. YouTube provides a facility to detect human faces and blur them. However, our analysis indicates that it fails with side view faces, occlusions, and low-resolution videos.

2 Components of a Redaction System

A typical redaction system relies on two key components: object detection and tracking. Object detection is required for automatic localization of relevant objects in a scene or video. Such automated localization prevents requiring manual tagging of the objects of interest. A tracking module then uses the tagged object information to estimate object positions in the subsequent frames. The performance of the detection and tracking modules control the efficiency of the redaction system—higher accuracy requires less manual review and/or validation of results. We review common detection and tracking techniques along with relevant datasets.

2.1 *Object Detection*

In the field of computer vision, object detection encompasses detecting the presence of and localizing objects of interest within an image. Object detection can assist the redaction problem by finding all of a certain category of object, for example faces, in a given image. The output of an object detection algorithm is typically a rectangular bounding box that encloses each object or a pixel-level segmentation of each object from its surroundings.

A variety of methods exist for object recognition and localization on still images (e.g., single video frames). A brief overview of some recent state-of-the-art techniques relevant to redaction is presented in the sections below. In Sec. 2.2, the ability to leverage temporal information to extend the results from object detection across a video sequence (i.e., series of video frames) is covered.

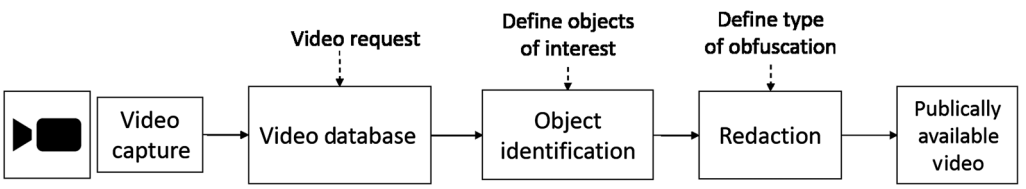


Fig. 2 Example of a typical video redaction system.

2.1.1 *Sliding window approach*

When considering the output of an object detection algorithm as a bounding box around the object(s) of interest, one intuitive method that has been applied is a sliding window approach. Here, a detection template is swept across the image, and at each location the response of an operation such as an inner product with the template or a more complex image classification is computed. The resulting detections (desired bounding box locations) are then selected as the template locations (center and template size) that meet a predetermined response threshold. For example, the Viola–Jones sliding-window method[10] was considered the state-of-the-art in face detection for some period of time. Unfortunately, sliding window-based approaches are computationally expensive as the number of windows can be very large to detect objects of different scale and sizes. As such, sliding window approaches are less common for video-based redaction applications.

2.1.2 *Region proposals*

Region proposals are candidate object subregions (windows) in an image that are computed based on low-level image features. A variety of studies[11–17] have suggested different region proposal generation methods using techniques such as hierarchical grouping, superpixel features, and graph cuts to score candidate region proposals. In most cases, region-proposal-based methods tend to oversample the image. Producing more candidate regions than actual objects reduces the likelihood of missing an object of interest (i.e., trades off more false alarms for fewer missed detections). Improved localization performance is then often achieved by pruning the raw set of candidate regions using some form of image-based classification step.

Object detection using the region proposals is generally based on a classifier. The classifier [e.g., neural networks, support vector machine (SVM), *k*-nearest neighbor, etc.] is applied to the features [e.g., CNN, Harris corners, SIFT, histogram of oriented gradient (HOG), etc.] extracted for each of the candidate region proposals to obtain a confidence score or probability for each candidate region. In alternate approaches, the model directly regresses a refined set of bounding box parameters (e.g., location, width, and height) for the final set of object detections.

2.1.3 *Recent methods*

Deep learning has achieved state-of-the-art results in a variety of different classification tasks in computer vision.[18] In 2014, Region Convolutional Neural Network (RCNN)[19] first applied the Convolutional Neural Network (CNN) architecture to the task of object localization. Since then, a variety of other deep learning-based methodologies for addressing the object detection and localization problem have been published, including fast-RCNN[20] and faster-RCNN.[21]

More recently, the you only look once (YOLO)[22] architecture was shown to achieve computational throughput speeds compatible with real-time video processing while also producing object localization results comparable with the prior methods. YOLO formulates object detection as a regression problem with the objective to predict bounding box coordinates and class confidence in an image by applying a single-pass CNN architecture. For redaction applications, the class labels and bounding box coordinates can be used to determine which image subregions should or should not be obfuscated.

2.1.4 *Pixel-based methods*

For redaction purposes, the output of an object detection might need to be inferred at a scale finer than a bounding box. For instance, consider the scenario in which there is PII for a bystander near a suspect's face in an image. Here, choosing to not redact a bounding box region around the suspect's face might be insufficient, allowing unwanted PII to be visible. An example illustrating the deficiency of using just a bounding box region around the suspect is shown in Fig. 3.

As an alternative to a bounding box approach, it is possible to assign a class label to each pixel in an image. Although similar learning-based algorithmic techniques as described above can be used, the output is a semantically segmented image. Recent approaches[24–27] based on fully convolutional neural network (FCN) methods[24,28] take in arbitrary size images and output region level classification for simultaneous detection and classification. Chen et al.[24] used conditional random fields to fine tune the fully convolutional output. Wu et al.[29] did extensive experiments to find optimum size and number of layers, then used bootstrapping and dropout for highly accurate segmentation.

For redaction applications, pixel-based methods for object localization can provide some advantages in terms of better differentiating foreground areas of interest (i.e., nonredacted regions) and surrounding background content that is to be redacted. However, any pixel-level missed detections in such a scenario translate into the potential for under-redaction of personal information. Thus, appropriate design choices must still be made to ensure acceptable overall redaction performance. Here, both over- and under-redactions must be appropriately considered. Appropriate performance measures for redaction will be discussed in more detail in Sec. 3.

2.2 *Object Tracking*

Similar to object detection in images, object tracking in videos is an important technology for automated redaction. Given the initial position and size of an object, a tracking algorithm should estimate the state of the object in subsequent

Fig. 3 Example of application of pixel-level segmentation on a sample image from AFLW[23] dataset.

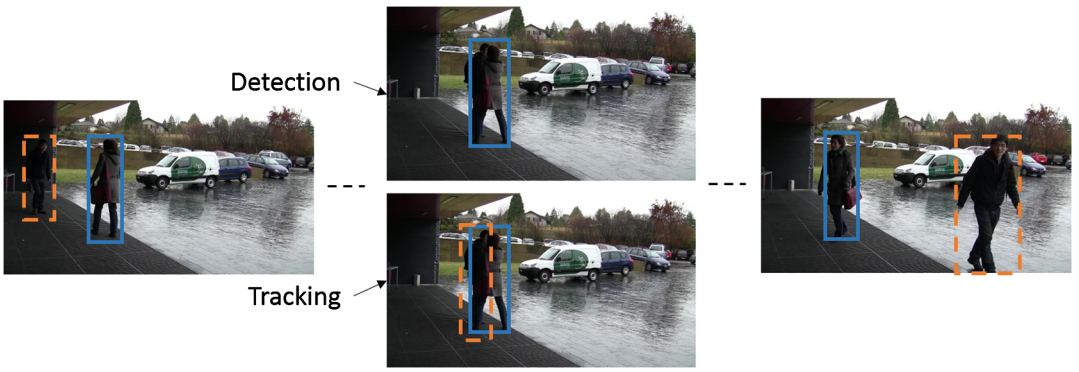


Fig. 4 Illustration of object detection compared with object tracking for a video stream. In the case of occlusions (center), tracking techniques are able to project the location of the object using temporal information. Example video from PEVid dataset.[30]

video frames. By maintaining a "lock" on the object of interest (person, face, license plate, etc.), the tracking algorithm helps to maintain object localization despite potential errors being committed by the object detector running on each video frame. An example of this is shown in Fig. 4.

Fundamentally, tracking an object involves extracting features of that object when first detected and then attempting to find similar (matching) features in subsequent frames. The major challenges associated with object tracking include illumination variation, changes in object scale, occlusions (partial or complete), changes in object pose or perspective relative to the camera, and motion blur. To be successful, tracking methods need to be robust to these types of noise factors, and tracking performance depends heavily on the features used.[31]

2.2.1 *Motion-model-based approaches*

Temporal differencing algorithms can detect objects in motion in the scene; alternatively, background subtraction, which requires the estimation of the stationary scene background, followed by subtraction of the estimated background from the current frame, can detect foreground objects (which include objects in motion). The output of either approach is a binary mask with the same pixel dimensions as the input video that has values equal to 0 where no motion/foreground objects are detected and values equal to 1 at pixel locations where motion/foreground objects are detected. This detection mask is usually postprocessed via morphological operations that discard detected objects with size and orientation outside predetermined ranges determined by the geometry of the image-capture scenario. Once candidate foreground objects have been detected, methods such as particle filtering are typically applied to leverage temporal information in linking objects in the current image frame with observations of these objects from prior frames.

2.2.2 *Appearance-based tracking*

Appearance-based trackers[32–34] rely on hand-crafted or machine-learned features of each object's visual appearance to isolate and match objects. Simple examples of this type of approach would include using fixed template matching (e.g., using two-dimensional correlation) or color histogram matching (e.g., using the mean-shift algorithm) to follow objects from frame-to-frame.

Appearance-based methods tend to be susceptible to large appearance changes due to varying illumination, heavy shadows, dramatic changes in perspective, etc. To address these issues, some approaches make use of adaptive color features[35] or online learned dictionaries of appearance models (such as in the track-learn-detect paradigm[36,37]) for objects that are being tracked. A key challenge with these types of methodologies then becomes the difficulty in tuning the online adaptation parameters. Here, the appearance models must be updated fast enough to accommodate changes in object appearance within the scene. However, tracker failures can increase if the models become overly responsive—incorporating too many extraneous appearance characteristics due to noise or surrounding background image content.

2.2.3 *Track-by-detection*

Detection of objects in individual frames can be extended to enable tracking and trajectory estimation. Recent success in object detection has led to development of tracking by detection[38–40] that uses CNN to track objects by detecting them in real time. Such approaches, however, have limitations in handling complex and long occlusions, where the fundamental object detection will struggle. A recent work on object detection in videos[41] used a temporal convolution network to predict multiple object locations and track the objects through a video.

Since the image-based detector must be applied at each video frame, a key challenge for track-by-detection methods tends to be the computational overhead required. However, since the YOLO approach leverages a single-shot network, it has been shown to achieve state-of-the-art object detection and localization in images at video frame rate speeds. Thus, YOLO is a natural candidate for tracking-by-detection-based approaches to following objects of interest in videos.

2.2.4 *Recurrent networks for object detection*

An object detector used for tracking performs frame-by-frame detection but fails to incorporate the temporal features present in the video. To overcome this, a recurrent neural network can be applied to exploit the history of object location.[40] The recurrent units in the form of long short term memory (LSTM)[42] cells use features from an object detector to learn temporal dependencies. The loss function for training the LSTM minimizes the error between the predicted and

ground truth bounding box coordinates. Although the utilization of temporal information by tracking across multiple frames can increase robustness, it significantly deteriorates the ability to perform tracking in real time.

In another recent method,[43] an online and offline tracker is proposed for multiobject tracking. Appearance (based on CNN features), shape, and motion of objects are used to compute the distance between current tracklets and obtained detections into an affinity matrix. The affinity is used as a measure to associate the current tracklets with the detections obtained in a frame using the Kuhn–Munkres algorithm. The offline tracker uses K-dense neighbors to associate tracklets and current detection.

2.3 *Datasets of Interest*

2.3.1 *Image datasets*

The most common objects redacted for privacy protection are faces, persons (i.e., full human body), house numbers, vehicle license plates, visible computer screens (which may be displaying PII), and skin regions or markings (e.g., tattoos). There are a number of published object detection datasets that are relevant to redaction. Although not tailored specifically to redaction, many of these datasets contain objects of interest for redaction. In addition, these data sets typically provide annotation of individual object locations and class labels, so they can easily support performance evaluation of redaction methods.

The widely used PASCAL visual object classes (VOC) dataset[44,45] has detailed semantic boundaries of 20 unique objects taken from consumer photos from the Flickr website. Among the object categories, the most relevant to redaction are the person and TV/monitor classes. However, the car, bus, bicycle, and motorbike classes could also be of interest depending on the redaction application. The complete dataset consists of 11,530 training and validation images. The "person" category is present in over 4000 images and the "tv/monitor" category in roughly 500 images.

Alternative datasets include an increased number of annotated categories and images. For example, MSCOCO objects[46] have 80 categories with 66,808 images having person and 1069 images having tv/monitor categories as pixel-level segmentations. The ImageNet objects[47] have 200 categories of common objects annotated with bounding boxes in over 450,000 images. The KITTI dataset[48] consists of 7481 training and 7518 test images collected from an autonomous driving setting and consists of annotated cars and pedestrians. Figure 5 shows sample images from the KITTI[48] and PASCAL datasets with annotated detection boxes and segmented pixels, respectively.

For exploring algorithms that specifically target the redaction of human faces, the annotated facial landmarks in the wild[23] (AFLW) is a popular dataset of $\sim 25,000$ annotated faces from real world images. This dataset includes variations of camera viewpoints, human poses, lighting conditions, and occlusions, all of which make face detection challenging. To make face detection more robust, Vu et al.[51] introduced a head detection dataset that contains 369,846 human heads annotated from movie frames, including difficult situations such as strong occlusions and low lighting conditions.

Face recognition datasets are also useful in redaction systems to test the efficacy of different obfuscation techniques. The LFW Face dataset[52] consists of 13,233 images of 5749 people collected from the internet. Similarly, the AT&T database[53] of faces contains 10 images each of 40 individuals. A more recent FaceScrub dataset[54] consists of 100,000 images of 530 celebrities.

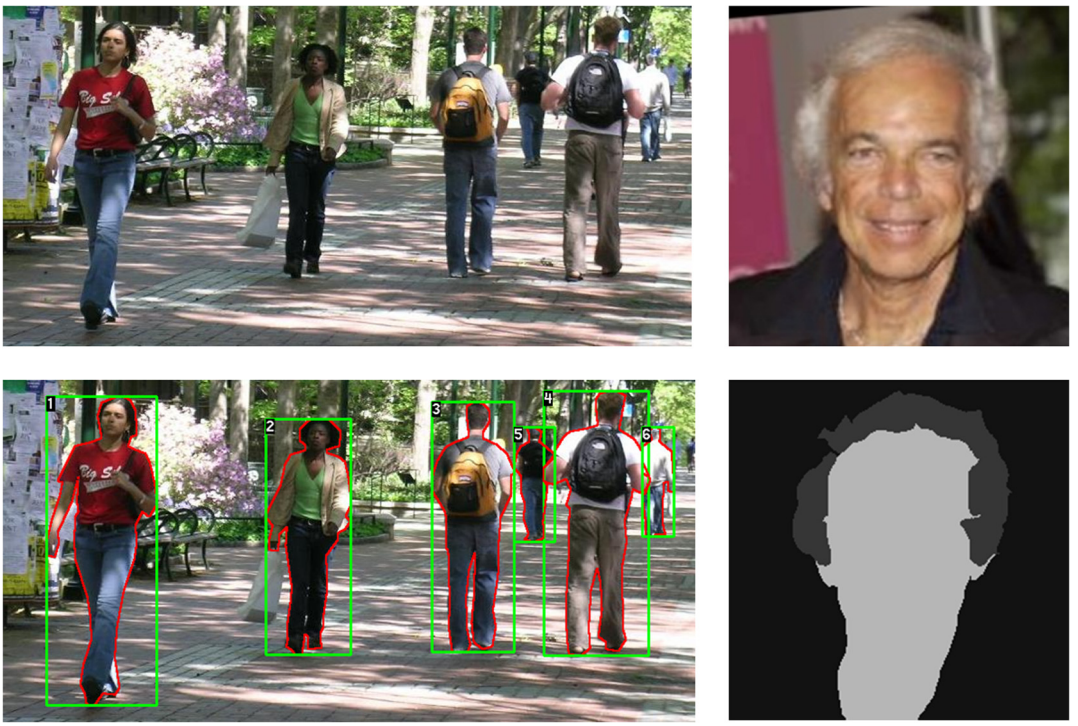


Fig. 5 Sample images from Penn-Fudan[49] and LFW[50] datasets with annotated detection boxes and segmented pixels.

2.3.2 *Video datasets*

Beyond still images, there are video datasets relevant to algorithm development and performance evaluation of redaction methods. One example is PEViD,[30] which was designed specifically with privacy and redaction-related issues in mind. The dataset consists of 21 clips of 16 s sampled at 25 fps at 1920×1080 resolution of surveillance videos in indoor and outdoor settings. The dataset has four different activities: walking, stealing, dropping, and fighting and has ground truth annotations for human body, face, body parts, and personal belongings such as bags. Another video dataset that can be useful to redaction is VIRAT,[55] which annotates vehicles and pedestrians on roadways.

In addition to detecting faces and heads, detection of the entire human body is very common in redaction. The Caltech Pedestrian[56] dataset is perhaps the most popular for detecting persons in images. It consists of ~10 h of 640×480 30 Hz video taken from a vehicle driving through regular traffic in an urban environment. About 250,000 frames (in 137 approximately minute long segments) with a total of 350,000 bounding boxes and 2300 unique pedestrians are annotated.

Object tracking in videos is a very popular task among researchers; hence, there are numerous datasets available to evaluate and benchmark tracking algorithms. The most common and recent ones are Object Tracking Benchmark (OTB),[57] YouTube Object dataset,[58] and ImageNet object detection from video.[59] In most datasets relevant to redaction, the target objects are humans or cars of small size in a surveillance-like setting with less motion in the background.

In current publicly available datasets, collections of annotated images are much more prevalent than video datasets. However, a common practice is to train on (large) image datasets and then apply the resulting models to video frames for evaluation (track by detection). Until the redaction-relevant, publically available video datasets increase substantially in size, this approach of training algorithms on available large image datasets will likely remain a key component of many redaction solutions.

3 Evaluation Metrics

The PASCAL VOC challenges[45] have been instrumental in setting forth standardized test procedures that enable fair comparison for benchmarking classification, object detection, and pixel segmentation algorithms. The questions "Is there a car in the image?," "Where are the cars located in the image?," and "Which pixels are devoted to cars?" are examples of classification, detection, and pixel segmentation problems. Scores are computed for each class and reported along with the average across all 21 classes. The PASCAL VOC challenges[45] use area under precision-recall curves. This is estimated at 11 equally spaced precision values from 0:0.1:1 to ensure that only methods with high recall across all precision values rank well.

Taking the detection task as an example, each object has a ground truth detection box. An automated method attempts to find each object and return a bounding box around the object. If the detected box precisely overlays the ground truth box, we have detected the object. But what do we do if the detected box is shifted up and to the left by a few pixels? How about shifted down and to the right by half the width of the object? PASCAL VOC utilizes the intersection over union (IOU) metric, whereby a bounding box is said to detect an object if IOU is >0.5. IOU is defined as

$$\mathrm{IOU} = \frac{\mu(\mathrm{GT_{BB}} \cap \mathrm{Det_{BB}})}{\mu(\mathrm{GT_{BB}} \cup \mathrm{Det_{BB}})} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN}}, \tag{1}$$

where μ is the set counting measure, which we define as area, $\mathrm{GT_{BB}}$ is the ground truth bounding box, $\mathrm{Det_{BB}}$ is the detected bounding box, and $\mu(\mathrm{GT_{BB}} \cap \mathrm{Det_{BB}})$ and $\mu(\mathrm{GT_{BB}} \cup \mathrm{Det_{BB}})$ are the areas of the intersection and union of the bounding boxes, respectively. TP = true positive (area of $\mathrm{Det_{BB}}$ that intersects $\mathrm{GT_{BB}}$), FP = false positive (area of $\mathrm{Det_{BB}}$ not intersected by $\mathrm{GT_{BB}}$), and FN = false negative (area of $\mathrm{GT_{BB}}$ not intersected by $\mathrm{Det_{BB}}$).

For a given application, FN and FP pixel detections can have different levels of importance. For instance, it is likely that redaction applications cannot tolerate many FN detections as personally identifying information (PII) may be exposed. Similarly, if a face is correctly obfuscated in all but a handful of frames, the video cannot be considered properly redacted. Likewise, the amount of tolerable FP can be dependent on the application. In some applications, it can be acceptable to blur a region slightly beyond a face. The acceptable amount of blurring beyond the face can depend on factors such as not wanting to, or conversely not having a concern for, obscuring neighboring information. Increasing FP tends to decrease FN. Taken to a limit, the entire frame can be obscured leading to zero FN, but very high FP. Once again, the acceptable levels of FN and FP will be highly dependent on the redaction application.

To enable optimization for a given problem, we first define normalized errors

$$\overline{\mathrm{FN}} = \frac{\mu(\mathrm{GT_{BB}} - \mathrm{Det_{BB}})}{\mu(\mathrm{GT_{BB}})}, \tag{2}$$

$$\overline{\mathrm{FP}} = \frac{\mu(\mathrm{Det_{BB}} - \mathrm{GT_{BB}})}{\mu(\mathrm{Det_{BB}})}, \tag{3}$$

where "−" denotes set subtraction.

These error measures can be extended by considering that certain pixels in a detection area can be more critical than others. For instance, pixels in the periocular region can be more useful in identifying a person than points farther out in a bounding box. Also, pixels in the bounding box but not directly on the object of interest can have a low level of importance. One approach to addressing critical pixels is with a saliency weighting. Let gt_i be the saliency weight of a pixel $x_i \in \mathrm{GT_{BB}}$. The saliency weighted $\overline{\mathrm{FN}}$ becomes the sum of the saliency weights in the missed pixels normalized by the sum of the saliency weights in the ground truth

$$\overline{\mathrm{FN}} = \frac{\sum_i gt_i \,\forall\, x_i \in \mu(\mathrm{GT_{BB}} - \mathrm{Det_{BB}})}{\sum_i p_i \,\forall\, x_i \in \mu(\mathrm{GT_{BB}})}. \tag{4}$$

Saliency can also be used to avoid over redaction or redacting objects that need to be viewed, such as a weapon. The saliency weights h_i would be for pixels in the image frame H but not in the ground truth bounding box $x_i \in (H - \mathrm{GT_{BB}})$. Saliency weighted false positive can be written as

Method	Case1	Case2	Case3	Case4	Case5	Case6	Case7
IOU	0.27	0.38	0.58	1.00	0.47	0.14	0.00
$\overline{\text{FN}}$	0.00	0.00	0.00	0.00	0.53	0.86	1.00
$1-\overline{\text{FN}}$	1.00	1.00	1.00	1.00	0.47	0.14	0.00
$\overline{\text{FP}}$	0.73	0.62	0.42	0.00	0.00	0.00	0.00
$1-\overline{\text{FP}}$	0.27	0.38	0.58	1.00	1.00	1.00	1.00
$\text{ACC}_R, \alpha = 0.5$	0.63	0.69	0.79	1.00	0.74	0.57	0.50
$\text{ACC}_R, \alpha = 0.75$	0.82	0.84	0.89	1.00	0.60	0.35	0.25

Fig. 6 Performance of IOU and ACC_R. The Det_{BB} goes from maximum $\overline{\text{FP}}$ to maximum $\overline{\text{FN}}$ in seven equal increments from left to right. Case 4 represents Det_{BB} that matches GT_{BB} exactly. The dotted green region represents GT_{BB}, and the blue dashed region represents Det_{BB}.

$$\overline{\text{FP}} = \frac{\sum_i h_i \,\forall\, x_i \in \mu(H - \text{Det}_{\text{BB}})}{\sum_i h_i \,\forall\, x_i \in \mu(H - \text{GT}_{\text{BB}})}. \tag{5}$$

This paper introduces the general concept of saliency for redaction; however, due to its application dependence, it is outside the scope of the paper to exercise it for various applications. Instead, we focus on unweighted $\overline{\text{FN}}$ and $\overline{\text{FP}}$ as per Eqs. (2) and (3).

To maintain similarity with the IOU metric that is prevalent in the field, we invert $\overline{\text{FN}}$ and $\overline{\text{FP}}$ errors to convert each into an accuracy, and then combine them into a single metric

$$\text{ACC}_R = \alpha(1-\overline{\text{FN}}) + (1-\alpha)(1-\overline{\text{FP}}). \tag{6}$$

Different values of α can be tailored to the specific applications. For example, in redaction, minimizing $\overline{\text{FN}}$ is generally more important than minimizing $\overline{\text{FP}}$; thus, $\alpha > 0.5$.

To demonstrate the applicability of ACC_R, Figs. 6 and 7 contain a few illustrative examples. The dotted green and blue dashed regions are s and Det_{BB}, respectively. The IOU row uses Eq. (1), and the ACC_R, $\alpha = 0.5$ and ACC_R, $\alpha = 0.75$ rows use Eq. (6). With regards to Fig. 6, on the left, the Det_{BB} of case 1 fills the entire image; as we step to the right, it occupies less and less until we get to case 4 where it occupies the area identical to GT_{BB}. As we continue moving to right, in case 7, Det_{BB} occupies zero area. The $\overline{\text{FN}}$ and $\overline{\text{FP}}$ rows show the false negative and positive errors, respectively, due to the mismatch between Det_{BB} and GT_{BB}. The ACC_R, $\alpha = 0.75$ row shows our recommended usage of Eq. (6) where, as desired for many redaction applications, false positives (on the left) are not penalized as much as false negatives (on the right).

With regards to Fig. 7, case 8 is a typical example, whereby the Det_{BB} is a mismatch to GT_{BB} and, in this case, is smaller than GT_{BB}. In case 8, by setting $\alpha = 0.75$, the $\overline{\text{FN}}$ error is penalized more heavily, and the $\overline{\text{FP}}$ error is penalized less. Since $\overline{\text{FP}} = 0$, the ACC_R, $\alpha = 0.75$ score is lower. In case 9, $\text{Det}_{\text{BB}} \subset \text{GT}_{\text{BB}}$ by the same amount as case 10 where $\text{GT}_{\text{BB}} \subset \text{Det}_{\text{BB}}$. As such, IOU and ACC_R, $\alpha = 0.5$ treat case 9 and case 10 the same. For obfuscation purposes, it is preferable to fully enclose the GT_{BB}. By weighting $\overline{\text{FN}} > \overline{\text{FP}}$, with $\alpha = 0.75$ in Eq. (6), the ACC_R, $\alpha = 0.75$ row of Fig. 7 shows the benefits of the introduced ACC_R metric. Similarly, by comparing case 11 with case 12, the ACC_R rows of Fig. 7 correctly report low values when Det_{BB} is much smaller or larger than GT_{BB}. Unlike IOU, which reports the same value for case 11 and case 12, ACC_R, $\alpha = 0.75$ clearly distinguishes the penalty for false negatives, which is how one might anticipate a redaction metric to behave.

To examine ACC_R further, Figs. 8 and 9 compare a sweep similar to Fig. 6. Figure 8 shows that the $\overline{\text{FN}}$ is zero until Det_{BB} becomes a subset of GT_{BB}. Similarly, $\overline{\text{FP}}$ is zero when $\text{Det}_{\text{BB}} \subset \text{GT}_{\text{BB}}$. By comparing IOU to ACC_R, $\alpha = 0.75$, we can see that $\overline{\text{FN}}$ is more important than $\overline{\text{FP}}$. Figure 9 examines the behavior of α in Eq. (6). When $\alpha = 0$, only the $\overline{\text{FP}}$ term in Eq. (6) is used, and it only penalizes ACC_R when $\text{GT}_{\text{BB}} \subset \text{Det}_{\text{BB}}$. Similarly, when $\alpha = 1$, only the $\overline{\text{FN}}$ term in Eq. (6) is used, and it only penalizes ACC_R when $\text{Det}_{\text{BB}} \subset \text{GT}_{\text{BB}}$. When $\alpha = 0.5$, ACC_R offers behavior similar to IOU and does not penalize false negatives appropriately for redaction applications. The solid blue line in Fig. 8 demonstrates our recommended $\alpha = 0.75$, appropriately favoring false positives over false negatives.

The above usage of Eq. (6) assumes that there is a single Det_{BB} and GT_{BB} per image. When multiple bounding boxes exist, all detection and ground truth boxes are merged into single, possibly fragmented, masks before applying Eq. (6). This ensures that all GT_{BB} regions are fully enclosed by one or more Det_{BB}. Using $\alpha = 0.75$ continues to penalize false negatives more than false positives.

Figure 10 compares the change in ACC_R with varying bounding box detections. We observe that, as the detected bounding box covers more ground truth area, i.e., the FN decreases, ACC_R becomes higher. As more area from the ground truth is missed, the ACC_R score is penalized. This is an important property of the redaction accuracy.

Method	Case8	Case9	Case10	Case11	Case12
IOU	0.66	0.38	0.38	0.11	0.11
$\overline{\text{FN}}$	0.34	0.62	0	0.89	0
$1-\overline{\text{FN}}$	0.66	0.38	1	0.11	1
$\overline{\text{FP}}$	0	0	0.62	0	0.89
$1-\overline{\text{FP}}$	1	1	0.38	1	0.11
$\text{ACC}_R, \alpha = 0.5$	0.83	0.69	0.69	0.56	0.56
$\text{ACC}_R, \alpha = 0.75$	0.74	0.54	0.85	0.33	0.78

Fig. 7 Performance of IOU and ACC_R on various test cases. The dotted green region represents GT_{BB}, and the dashed blue region represents Det_{BB}.

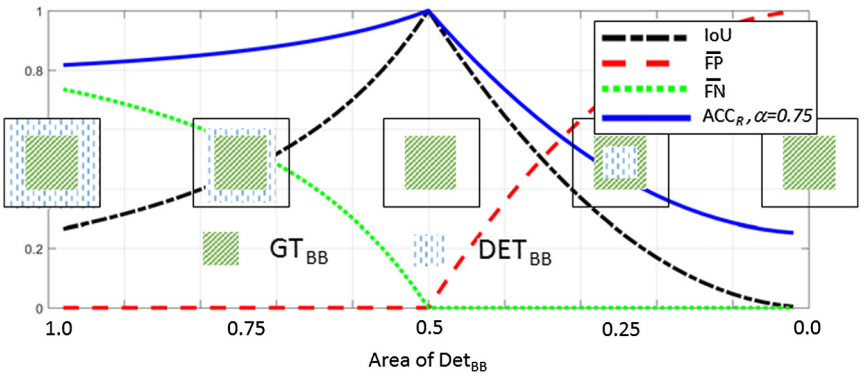


Fig. 8 Comparison of IOU, $\overline{\text{FN}}$, $\overline{\text{FP}}$, and ACC$_R$. GT$_{BB}$ is fixed at 50% of input image volume, and Det$_{BB}$ goes from maximum $\overline{\text{FP}}$ to maximum $\overline{\text{FN}}$ from left to right.

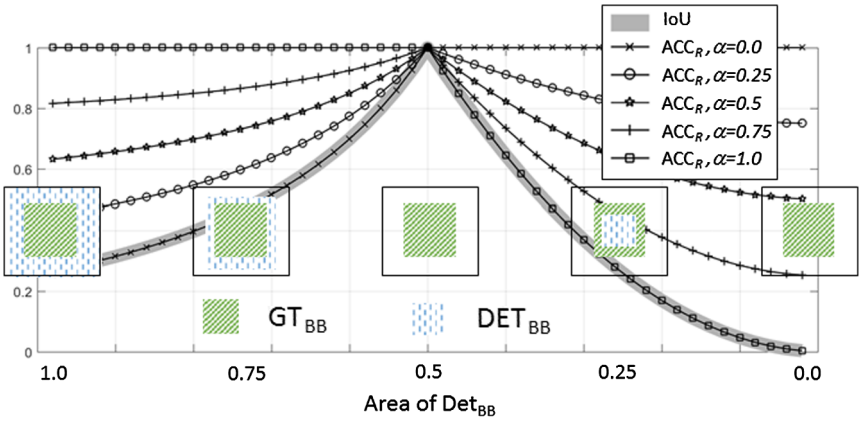


Fig. 9 Analysis of ACC$_R$. GT$_{BB}$ is fixed at 50% of input image volume, and Det$_{BB}$ goes from maximum $\overline{\text{FP}}$ to maximum $\overline{\text{FN}}$ from left to right.

3.1 *Comparison of Methods*

We evaluate the proposed metric on four different object categories that are relevant to redaction—faces, human heads, persons, and tv/monitors. For faces, we use the AFLW[23] dataset. The faster-RCNN technique uses 50% of images for training and the remaining 50% for testing. To compare recent deep learning methods with a classical object detector, we used the HOG feature combined with a linear classifier, an image pyramid, and sliding window detection scheme for the face and person categories. The mean redaction accuracy (mACC$_R$) is defined as the average score over all test images to compare the performance over a dataset.

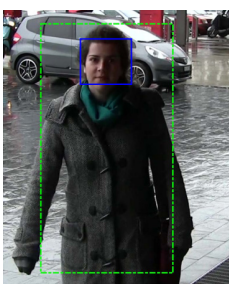
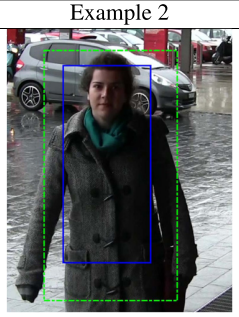
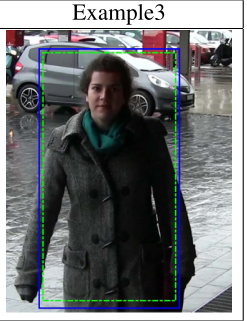
Metrics	Example 1	Example 2	Example3
			
IOU	0.070	0.518	0.905
$\overline{\text{FN}}$	0.93	0.42	0.00
$\overline{\text{FP}}$	0.00	0.00	0.091
$\text{ACC}_R(\alpha = 0.3)$	0.767	0.879	0.928
$\text{ACC}_R(\alpha = 0.5)$	0.535	0.759	0.952
$\text{ACC}_R((\alpha = 0.7)$	0.303	0.639	0.976

Fig. 10 Comparison of IOU and ACC$_R$ with varying values of α. Solid blue line is detected bounding box and dashed green line is ground truth bounding box. Example image from PEVid dataset.[30]

Table 1 Performance comparison for face detection using classical HOG features with a linear classifier and the more recent faster-RCNN method. $m\overline{FN}$ and $m\overline{FP}$ are mean normalized errors, $mACC_R$ is mean accuracy for redaction, and mAP is mean average precision.

Method	HOG + Lin. classifier	Faster-RCNN
$m\overline{FN}$	0.322	0.058
$m\overline{FP}$	0.089	0.338
mAP	0.364	0.672
$mACC_R(\alpha = 0.1)$	0.887	0.689
$mACC_R(\alpha = 0.3)$	0.840	0.745
$mACC_R(\alpha = 0.5)$	0.793	0.801
$mACC_R(\alpha = 0.7)$	0.747	0.857
$mACC_R(\alpha = 0.9)$	0.700	0.913

As reported in Table 1, the faster-RCNN method achieves lower FN but higher FP compared with the HOG-feature-based method. This is typically a desirable property in a redaction system where failing to redact parts of an object may reveal sensitive information. Analogously, faster-RCNN is advantaged for higher alpha values where FN results in a higher penalty than FP. Conversely, applications that are required to penalize FP more than FN would benefit from lower α values and the HOG-based method.

Since for a redaction application, missing side views or occluded faces can also reveal the identity of persons, we run experiments for comparison of face and head detectors. We train a head detector to compare the robustness in detecting faces due to occlusions and view angles. The YOLO model was trained on images from the Head Annotations[51] dataset. Testing was done on the FDDB face detection dataset[60] with 5171 faces in 2845 images, and the results are reported in Table 2. The testing was done on a dataset with ground truth only for faces (and not heads), and the qualitative improvement do not directly translate to the metrics. Therefore, we show examples in Fig. 11 comparing face and head detectors. The mAP scores are low due to cross dataset testing. The YOLO framework has strong spatial constraints imposed by limiting two boxes per grid cell; hence, it fails to detect small objects that appear in groups. Moreover, since the learning is done to predict bounding boxes from data, it struggles to generalize to objects in new or unusual aspect ratios or configurations.

Table 2 Performance comparison of YOLO model trained on face and head datasets for testing on a face dataset. $m\overline{FN}$ and $m\overline{FP}$ are mean normalized errors, $mACC_R$ is mean accuracy for redaction, and mAP is mean average precision.

Method/training data	YOLO/face	YOLO/head
$m\overline{FN}$	0.687	0.566
$m\overline{FP}$	0.138	0.172
mAP	0.090	0.142
$mACC_R(\alpha = 0.1)$	0.367	0.787
$mACC_R(\alpha = 0.3)$	0.477	0.709
$mACC_R(\alpha = 0.5)$	0.587	0.630
$mACC_R(\alpha = 0.7)$	0.696	0.551
$mACC_R(\alpha = 0.9)$	0.806	0.472

For the "person" and "tv/monitor" object categories, we used the standard train/test splits from the PASCAl-VOC 2007[45] dataset. The results are reported in Tables 3 and 4. The faster-RCNN achieved the lowest false negatives and hence the best mAP scores among the three methods. Among different techniques, the selection is based on the method that is best suited to detect an object category. Similarly, the selection of α values depends on the desired performance in terms of FN versus FP.

3.2 *Tracking Results*

Since there may be high costs (e.g., lawsuits) associated with releasing improperly redacted videos, some degree of human review and validation is typically required. In semiautomated schemes, confidence scores from the automated redaction system are typically used to determine when a manual review by a skilled technician is needed on particular video frames. Because manual review and editing is costly, it is desirable for the redaction system to have a low percentage of missed (low confidence) frames.

Evaluation of object tracking in videos is done using a threshold on ACC_R to obtain the percentage of missed frames as reported in Table 5. The performance of recent object tracking models is evaluated on a subset of the OTB[57] dataset. We report results using a correlation filter-based tracker implemented using *DLib*[61] and compare it with a more recent multidomain trained CNN-based object tracker.[62] The first frame of each video is manually tagged to initialize the correlation tracker, and its performance showed minimal changes. This may be due to the complexity of the videos and simplicity of the tracker. With sufficient training data, recent deep learning-based techniques can achieve high accuracies and reduce the amount of manual intervention required in video redaction systems. The α value controls the contribution of FN and FP in the accuracy score. For example, the variation in ACC_R values with α for the MDNet method indicates that it has higher FN than FP. While designing a redaction system, the threshold on the accuracy would determine the number of frames that require a manual review. This also depends on a number of other factors such as the object of interest, tracking method, and desired performance in protecting the object (FP versus FN).

4 Types of Obfuscation

Once all objects with private information are detected, the information needs to be obfuscated in a manner that protects the privacy. These obfuscations can be simple methods such as blanking or masking objects such as faces with shapes in individual video frames. Other common obfuscations are blurring, pixelation, or interpolation with the surroundings.

Fig. 11 Examples to qualitatively compare a face and head detector. (a) Original images, (b) YOLO face, and (c) head detector outputs, where the bounding boxes are outputs of the YOLO models. Example images from the AFLW dataset.[23]

Table 3 Performance comparison for person detection using classical HOG features with a linear classifier and more recent faster-RCNN and YOLO methods. m$\overline{\text{FN}}$ and m$\overline{\text{FP}}$ are mean normalized errors, mACC_R is mean accuracy for redaction, and mAP is mean average precision.

Method	HOG + Lin. classifier	Faster-RCNN	YOLO
m$\overline{\text{FN}}$	0.415	0.119	0.189
m$\overline{\text{FP}}$	0.372	0.220	0.202
mAP	0.194	0.758	0.654
$\text{mACC}_R(\alpha = 0.1)$	0.057	0.776	0.735
$\text{mACC}_R(\alpha = 0.3)$	0.048	0.796	0.752
$\text{mACC}_R(\alpha = 0.5)$	0.039	0.816	0.769
$\text{mACC}_R(\alpha = 0.7)$	0.031	0.836	0.786
$\text{mACC}_R(\alpha = 0.9)$	0.022	0.856	0.803

Table 4 Performance comparison for tv/monitor detection using faster-RCNN and YOLO methods. m$\overline{\text{FN}}$ and m$\overline{\text{FP}}$ are mean normalized errors, mACC_R is mean accuracy for redaction, and mAP is mean average precision.

Method	Faster-RCNN	YOLO
m$\overline{\text{FN}}$	0.159	0.344
m$\overline{\text{FP}}$	0.404	0.189
mAP	0.661	0.669
$\text{mACC}_R(\alpha = 0.1)$	0.588	0.624
$\text{mACC}_R(\alpha = 0.3)$	0.637	0.638
$\text{mACC}_R(\alpha = 0.5)$	0.686	0.652
$\text{mACC}_R(\alpha = 0.7)$	0.735	0.666
$\text{mACC}_R(\alpha = 0.9)$	0.784	0.679

More complex methods include geometric distortion and scrambling that allows decryption with a key. We discuss common obfuscation methods below. Several examples are shown in Fig. 12.

First, consider various approaches that can be taken for bounding the region to be obscured using blurring as an example obscuration method. At a coarse level, an entire image frame that contains any sensitive information can be blurred. This may be useful when the video is relatively long compared with a small number of frames that need to be obscured or it is determined that blurred information is sufficient for the viewer. For instance, in a courtroom showing an auto accident, the overall movement of the vehicles may be adequately observed in a video that is blurred to a degree that obscures the identify of persons and license plates in the video. Blurring the entire frame is a simple method for protecting information and can ensure a high level of protection, but important context of the scene may get lost.

The sensitive region can be defined as the detected bounding box around the subject of interest. The tolerable "looseness" of the bounding box balances the trade-off between

Table 5 Comparison of percentage of missed frames on a subset of the OTB object tracking dataset.[57]

Method		Correlation tracker[61]		MDNet[62]	
α	Threshold ACC_R	$mACC_R$	% of missed frames	$mACC_R$	% of missed frames
0.3	0.5	0.474	45.90	0.85	0.279
0.5		0.473	45.95	0.819	0.619
0.7		0.472	46.09	0.787	8.64
0.3	0.7	0.474	46.27	0.85	1.14
0.5		0.473	47.05	0.819	14.43
0.7		0.472	50.09	0.787	26.12

false positives (which can obscure context) and false negatives (which potentially reveal PII). A looser bounding box increases FP and decreases FN, and vice versa. This trade-off should be selectable according to the given application requirements. If the general shape of the sensitive information is known, the detection box can be used to place an alternative mask in that region. For example, ellipses of different aspect ratios are sometimes used for face and body redaction. As indicated in Sec. 2.1.4, object detection might need to be inferred at a scale finer than a bounding box.

Obscuration methods must be understood in the context of their ability to suitably mask the sensitive information and the parameters used within the method. Fully blanking out or masking pixels in the sensitive region is the most secure method, but this can significantly affect certain contextual information, such as movement and actions in the region. More typical is blurring using a Gaussian blur kernel, which brings in the issue of selecting Gaussian parameters that provide adequate obfuscation. Pixelation (mosaicing) is another common obfuscation method. The region to be obfuscated is partitioned into a square grid, and the average color of the pixels contained within each square is computed and used for all pixels within the square. Increasing the size of the squares increases the level of obfuscation. Figure 13 shows example images with varying degrees of blurring and pixelation. Blurring was performed using the Gaussian blurring function in OpenCV.[63] The standard deviation of the Gaussian kernel was varied to achieve multiple degrees of blurring. The degree of pixelation was controlled by changing the size of the squares used in averaging.

Interpolation[66] with the background can be useful in applications that require the blurred image to be free from redaction artifacts. Studies such as Ref. 67 have also studied skin-color-based face detection.

In some applications, there is a requirement to retrieve the original object after redaction. This can allow release of the video where authorized parties possess a key that enables decryption. A system to retrieve the original data with proper authentication is presented by Cheung et al.[68] They use a rate-distortion optimized data-hiding scheme using an RSA key that allows access only to authenticated individuals. Similarly, Ref. 69 presented a retrievable object obscuring method.

Similar to the visual content, audio is also an integral part of surveillance videos. Detecting the audio segment to redact can either be based on the object detection in the parallel video stream or can be an independent search for audio clips. The audio segment could be replaced with a beep, muted, or modulated such that the original sound is protected.

4.1 *Recognition in Obfuscated Images*

The degree of obfuscation is an important consideration in the prevention of unwanted identification of redacted faces or objects. In fact, under constrained conditions, a fairly accurate face recognition can be achieved given some prior knowledge of the blur kernel and obfuscation technique.[70] Recently, McPherson et al.[71] studied the limitations faced by ad-hoc image obfuscation techniques. They trained deep convolution network classifiers on obfuscated images. Their results show that faces or objects can be recognized using trained models even if the image has been obfuscated with high levels of pixelation, blurring, or encrypting the significant JPEG components. Other studies have also reported techniques and results on recognition of blurred faces.[72,73] Chen at al.[5] presented a study in face masking and showed that face-masked images have a chance of exposing a person's identity through a pairwise attack. They presented a technique to obscure the entire body and claimed that it has better potential for privacy protection than face-masking.

Collectively, these results indicate that care must be taken when designing the obfuscation component of a redaction system. Parameters of the method should be chosen to assure acceptable, low levels of reidentification accuracy using

Fig. 12 Illustration of various obfuscation techniques. (a) Original image, (b) full frame blurred, (c) bounding box of face blurred, (d) bounding box of face blanked out, and (e) pixelated bounding box of face.

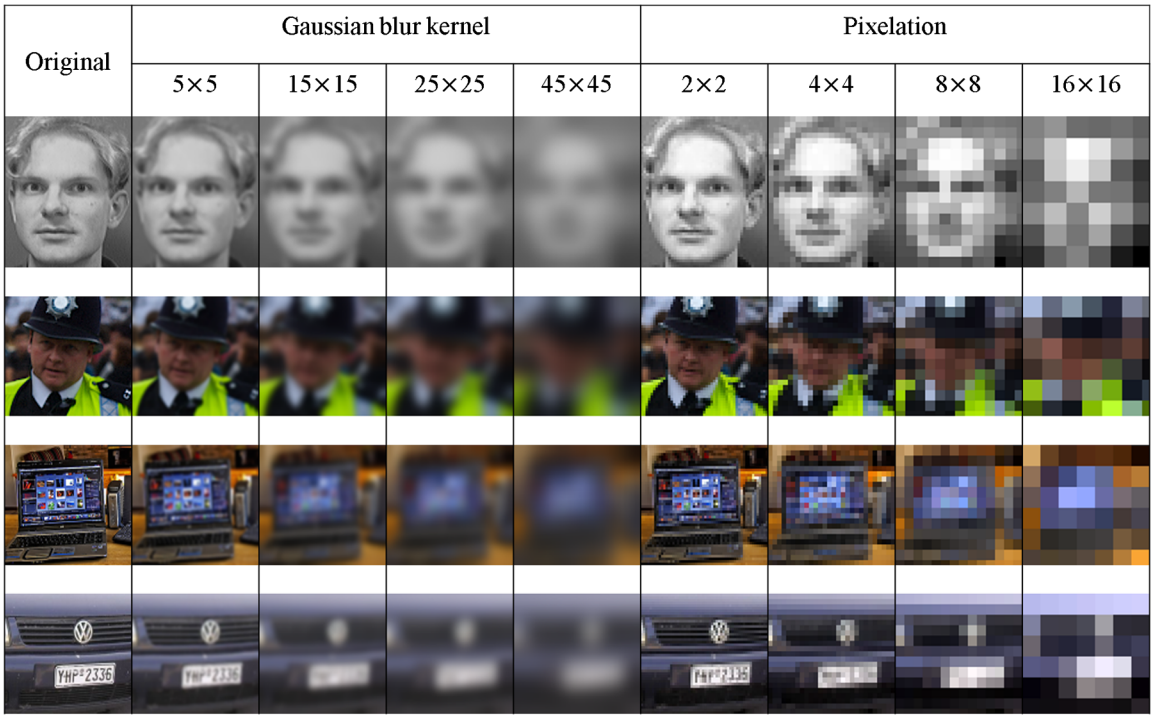


Fig. 13 Varying degrees for obfuscation on example images from (top to bottom) AT&T,[64] AFLW,[23] VOC,[45] and MediaLab LPR[65] datasets. All images are cropped to dimensions 92×112. Standard deviation of Gaussian kernel is varied as 5×5, 15×15, 25×25, and 45×45. Pixelation window size is varied as 2×2, 4×4, 8×8, and 16×16.

Table 6 Comparison of face recognition accuracy using an SVM classifier by varying degrees of blurring and pixelation on the AT&T face dataset.

		Gaussian kernel				Pixelation			
	Orig.	5×5	15×15	25×25	45×45	2×2	4×4	8×8	16×16
Top 1 accuracy	88.74	85.25	73.75	61.25	31.25	87.5	83.75	70.0	36.25

known techniques. To further illustrate this point, we provide an example of face recognition from images with varying degrees of obfuscation. We use the AT&T database of faces,[64] which consists of 10 different images of dimensions 92×112 pixels, each of 40 distinct subjects. These include images taken at different times, with variations in lighting, facial expressions, and facial details. The results for face recognition are reported in Table 6. For each subject, eight images were used for training and two for testing. An SVM classifier was trained on the top 150 Eigenfaces (principal component analysis) of the unlabeled training dataset.

The results of this experiment provide empirical evidence that it becomes increasingly difficult to recognize redacted faces as the degree of obfuscation is increased. This is true even under conditions where the exact redaction method applied is known *a priori* and where the identification task is to select the most similar individual from a small pool of candidates (versus a database of thousands or millions of people).

5 Open Problems

We discuss several open problems and challenges associated with video redaction systems.

Although state-of-the-art computer vision is increasingly robust in detecting certain objects, such as faces, bodies, and license plates, the sensitive PII can take on many diverse forms that will confound attempts to fully automate the process (e.g., skin, tattoos, house numbers written in script, logos, store front signs, street signs, and graffiti). Skin occurs in many tones, and color-based segmentation is not robust for sensitive applications. While character recognition may be robust for conventional documents, recognition in the outdoors is a different problem. The video may have been captured in very suboptimal conditions, such as poor lighting and geometric perspective. In any given application, particular objects may need to be obfuscated while other instances of that object class must be clearly visible (blur face 1 but not face 2).

The public concern over privacy coupled with the need for low cost ever vigilant security will drive privacy protection into smart cameras, so certain material is never stored or transmitted, except possibly with special encryption. While some complex custom obscurations may not be possible, mainline tasks such as face obscuration could be performed by computing on the edge. The performance of such redaction systems would depend on the accuracy of the face

detection and obfuscation methods. Moreover, the processing time becomes a critical requirement since the amount of data is ever increasing.

Law enforcement applications cannot release any sensitive data. If a single frame in a video is missed by a redaction system, it could reveal the identity, for example, of a witness and put them in danger. That one missed frame can defeat the value of redacting thousands of other frames in the video sequence. This sensitivity necessitates a manual review of the redacted output. Efficient review methods can greatly reduce labor costs.

6 Conclusion

With the rising popularity of surveillance, body, car, and cell phone recording devices, imagery is increasingly being used for public purposes such as law enforcement, criminal courts, and news services. Often, the personal identity of people, their cars, businesses, or homes are identifiable in these recordings. Video redaction or obfuscation of personal information in videos for privacy protection is becoming very important. Object detection and tracking are two key components of a redaction system. The current advances in the field of deep learning achieve state-of-the-art performances in object detection and tracking. However, the current evaluation metrics do not consider redaction-specific constraints. The presented redaction metric is promising for evaluating redaction systems. We compare classical methods with recent deep learning-based methods on redaction-specific object categories. While designing a video redaction system, the most desired property is having a fewer number of frames that require a manual review. This depends on factors such as threshold on the accuracy, the object of interest, detection and tracking method, and desired performance in protecting the object (FP versus FN). More challenges such as processing time, raw video retrieval, and manual review remain active research areas.

References

1. N. Jenkins, "245 million video surveillance cameras installed globally in 2014," IHS Markit, https://technology.ihs.com/532501/245-million-video-surveillance-cameras-installed-globally-in-2014 (11 June 2015).
2. B. A. Reaves, "Local police departments, 2013: equipment and technology," Bureau of Justice Statistics, https://www.bjs.gov/content/pub/pdf/lpd13et.pdf (9 February 2016).
3. "Annual report, 2015," Technical Report, Major Cities Chiefs Association, https://www.majorcitieschiefs.com/pdf/news/annual_report_2015.pdf (3 March 2016).
4. J. Schiff et al., "Respectful cameras: detecting visual markers in real-time to address privacy concerns," in *Protecting Privacy in Video Surveillance*, A. Senior, Ed., pp. 65–89, Springer, London (2009).
5. D. Chen et al., "Protecting personal identification in video," in *Protecting Privacy in Video Surveillance*, A. Senior, Ed., pp. 115–128, Springer, London (2009).
6. A. Pande and J. Zambreno, *Securing Multimedia Content Using Joint Compression and Encryption*, pp. 23–30, Springer, London (2013).
7. P. Korshunov and T. Ebrahimi, "Using warping for privacy protection in video surveillance," in *18th Int. Conf. on Digital Signal Processing (DSP)*, pp. 1–6 (2013).
8. J. Wickramasuriya et al., "Privacy protecting data collection in media spaces," in *Proc. of the 12th Annual ACM Int. Conf. on Multimedia*, pp. 48–55, ACM (2004).
9. J. J. Corso et al., "Video analysis for body-worn cameras in law enforcement," arXiv preprint arXiv:1604.03130 (2016).
10. P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. of IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR 2001)*, Vol. 1, pp. I-511–I-518 (2001).
11. C. Gu et al., "Recognition using regions," in *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 1030–1037 (2009).
12. J. Carreira and C. Sminchisescu, "Constrained parametric min-cuts for automatic object segmentation," in *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pp. 3241–3248 (2010).
13. P. Arbeláez et al., "Multiscale combinatorial grouping," in *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 328–335 (2014).
14. I. Endres and D. Hoiem, "Category independent object proposals," in *European Conf. on Computer Vision*, pp. 575–588, Springer, Berlin, Heidelberg (2010).
15. J. R. R. Uijlings et al., "Selective search for object recognition," *Int. J. Comput. Vision* **104**(2), 154–171 (2013).
16. M. M. Cheng et al., "Bing: binarized normed gradients for objectness estimation at 300 fps," in *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 3286–3293 (2014).
17. C. L. Zitnick and P. Dollár, "Edge boxes: locating object proposals from edges," in *European Conf. on Computer Vision*, pp. 391–405 (2014).
18. C. Szegedy et al., "Going deeper with convolutions," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 1–9 (2015).
19. R. Girshick et al., "Rich feature hierarchies for accurate object detection and semantic segmentation," in *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 580–587 (2014).
20. R. Girshick, "Fast R-CNN," in *IEEE Int. Conf. on Computer Vision (ICCV)*, pp. 1440–1448 (2015).
21. S. Ren et al., "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.* **39**, 1137–1149 (2017).
22. J. Redmon et al., "You only look once: unified, real-time object detection," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788 (2016).
23. M. Köstinger et al., "Annotated facial landmarks in the wild: a large-scale, real-world database for facial landmark localization," in *IEEE Int. Conf. on Computer Vision Workshops (ICCV Workshops)*, pp. 2144–2151 (2011).
24. L.-C. Chen et al., "Deeplab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," arXiv preprint arXiv:1606.00915 (2016).
25. Z. Liu et al., "Semantic image segmentation via deep parsing network," in *IEEE Int. Conf. on Computer Vision (ICCV)*, pp. 1377–1385 (2015).
26. H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *IEEE Int. Conf. on Computer Vision (ICCV)*, pp. 1520–1528 (2015).
27. G. Lin et al., "Efficient piecewise training of deep structured models for semantic segmentation," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 3194–3203 (2016).
28. J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 3431–3440 (2015).
29. Z. Wu, C. Shen, and A. V. D. Hengel, "High-performance semantic segmentation using very deep fully convolutional networks," arXiv preprint arXiv:1604.04339 (2016).
30. P. Korshunov and T. Ebrahimi, "PEViD: privacy evaluation video dataset," *Proc. SPIE* **8856**, 88561S (2013).
31. N. Wang et al., "Understanding and diagnosing visual tracking systems," in *IEEE Int. Conf. on Computer Vision (ICCV)*, pp. 3101–3109 (2015).
32. T. B. Dinh, N. Vo, and G. Medioni, "Context tracker: exploring supporters and distracters in unconstrained environments," in *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 1177–1184 (2011).
33. J. A. F. Henriques et al., "Exploiting the circulant structure of tracking-by-detection with kernels," in *Proc. of the 12th European Conf. on Computer Vision*, Vol. Part IV, pp. 702–715 (2012).
34. J. Kwon and K. M. Lee, "Visual tracking decomposition," in *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pp. 1269–1276 (2010).
35. M. Danelljan et al., "Adaptive color attributes for real-time visual tracking," in *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 1090–1097 (2014).
36. Z. Kalal, J. Matas, and K. Mikolajczyk, "Online learning of robust object detectors during unstable tracking," in *IEEE 12th Int. Conf. on Computer Vision Workshops (ICCV Workshops)*, pp. 1417–1424 (2009).
37. Z. Kalal, K. Mikolajczyk, and J. Matas, "Face-TLD: tracking-learning-detection applied to faces," in *IEEE Int. Conf. on Image Processing*, pp. 3789–3792 (2010).
38. L. Wang et al., "STCT: sequentially training convolutional networks for visual tracking," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 1373–1381 (2016).
39. F. K. Becker et al., "Automatic equalization for digital communication," *Proc. IEEE* **53**, 96–97 (1965).
40. G. Ning et al., "Spatially supervised recurrent convolutional neural networks for visual object tracking," arXiv preprint arXiv:1607.05781 (2016).
41. K. Kang et al., "Object detection from video tubelets with convolutional neural networks," arXiv preprint arXiv:1604.04053 (2016).
42. S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.* **9**(8), 1735–1780 (1997).

43. F. Yu et al., "POI: multiple object tracking with high performance detection and appearance feature," *European Conf. on Computer Vision*, pp. 36–42 (2016).
44. M. Everingham et al., "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vision* **88**(2), 303–338 (2010).
45. M. Everingham et al., "The Pascal visual object classes challenge: a retrospective," *Int. J. Comput. Vision* **111**(1), 98–136 (2015).
46. T.-Y. Lin et al., "Microsoft COCO: common objects in context," in *Proc. of the 13th European Conf. on Computer Vision*, Vol. Part IV, pp. 740–755 (2014).
47. J. Deng et al., "ImageNet: a large-scale hierarchical image database," in *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 248–255 (2009).
48. A. Geiger et al., "Vision meets robotics: the KITTI dataset," *Int. J. Rob. Res.* **32**(11), 1231–1237 (2013).
49. L. Wang et al., "Object detection combining recognition and segmentation," in *Proc. of the 8th Asian Conf. on Computer Vision*, Vol. Part I, pp. 189–199, Springer-Verlag, Berlin, Heidelberg (2007).
50. A. Kae et al., "Augmenting CRFs with Boltzmann machine shape priors for image labeling," in *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 2019–2026 (2013).
51. T. H. Vu, A. Osokin, and I. Laptev, "Context-aware CNNs for person head detection," in *IEEE Int. Conf. on Computer Vision (ICCV)*, pp. 2893–2901 (2015).
52. G. B. Huang et al., "Labeled faces in the wild: a database for studying face recognition in unconstrained environments," Technical Report 07-49, University of Massachusetts, Amherst (2007).
53. F. S. Samaria and A. C. Harter, "The database of faces," AT&T-Laboratories-Cambridge, http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html (11 June 2015).
54. H. W. Ng and S. Winkler, "A data-driven approach to cleaning large face datasets," in *IEEE Int. Conf. on Image Processing (ICIP)*, pp. 343–347 (2014).
55. S. Oh et al., "A large-scale benchmark dataset for event recognition in surveillance video," in *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 3153–3160 (2011).
56. P. Dollar et al., "Pedestrian detection: a benchmark," in *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 304–311 (2009).
57. Y. Wu, J. Lim, and M.-H. Yang, "Object tracking benchmark," *IEEE Trans. Pattern Anal. Mach. Intell.* **37**(9), 1834–1848 (2015).
58. A. Prest et al., "Learning object class detectors from weakly annotated video," in *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 3282–3289 (2012).
59. O. Russakovsky et al., "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vision* **115**(3), 211–252 (2015).
60. V. Jain and E. G. Learned-Miller, "FDDB: a benchmark for face detection in unconstrained settings," Technical Report, University of Massachusetts, Amherst (2010).
61. D. E. King, "Dlib-ml: a machine learning toolkit," *J. Mach. Learn. Res.* **10**, 1755–1758 (2009).
62. H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 4293–4302 (2016).
63. G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision with the OpenCV Library*, O'Reilly Media, Inc., Sebastopol (2008).
64. F. S. Samaria and A. C. Harter, "Parameterisation of a stochastic model for human face identification," in *Proc. of IEEE Workshop on Applications of Computer Vision*, pp. 138–142 (1994).
65. I. Anagnostopoulos et al., "License plate recognition from still images and video sequences: A survey," *IEEE Trans. Intell. Transp. Syst.* **9**(3), 377–391 (2008).
66. J. Wickramasuriya et al., "Privacy-protecting video surveillance," *Proc. SPIE* **5671**, 64 (2005).
67. S. K. Singh et al., "A robust skin color based face detection algorithm," *Tamkang J. Sci. Eng.* **6**(4), 227–234 (2003).
68. S.-C. Cheung et al., "Protecting and managing privacy information in video surveillance systems," in *Protecting Privacy in Video Surveillance*, A. Senior, Ed., pp. 11–33, Springer, London (2009).
69. F. Dufaux and T. Ebrahimi, "Scrambling for privacy protection in video surveillance systems," *IEEE Trans. Circuits Syst. Video Technol.* **18**(8), 1168–1174 (2008).
70. P. Vageeswaran, K. Mitra, and R. Chellappa, "Blur and illumination robust face recognition via set-theoretic characterization," *IEEE Trans. Image Process.* **22**, 1362–1372 (2013).
71. R. McPherson, R. Shokri, and V. Shmatikov, "Defeating image obfuscation with deep learning," arXiv preprint arXiv:1609.00408 (2016).
72. V. Ojansivu and J. Heikkilä, "Blur insensitive texture classification using local phase quantization," in *Proc. of the 3rd Int. Conf. on Image and Signal Processing (ICISP, '08)*, pp. 236–243, Springer-Verlag, Berlin, Heidelberg (2008).
73. A. Hadid, M. Nishiyama, and Y. Sato, "Recognition of blurred faces via facial deblurring combined with blur-tolerant descriptors," in *20th Int. Conf. on Pattern Recognition*, pp. 1160–1163 (2010).

Shagan Sah obtained his bachelors in engineering from the University of Pune, India and his MS degree in imaging science from Rochester Institute of Technology (RIT), USA with aid of RIT Graduate Scholarship. He is currently a PhD candidate in the Center for Imaging Science at RIT. His current interests lie in the intersection of machine learning, natural language processing and computer vision for image and video understanding. He has worked at Motorola, Xerox-PARC and Cisco Systems.

Ameya Shringi is a master's student in B. Thomas Golisano College of Computing and Information Sciences and a member of Machine Intelligence Laboratory at RIT, NY, USA. He graduated from Vellore Institute of Technology in 2011 with a Bachelor of Technology and has worked with Kitware Inc. His research interests include applications of machine learning models for object tracking in surveillance videos.

Raymond Ptucha is an assistant professor in computer engineering and director of the Machine Intelligence Laboratory at Rochester Institute of Technology. His research specializes in machine learning, computer vision, and robotics. He graduated from RIT with MS degree in image science and PhD in computer science. He is a passionate supporter of STEM education and is an active member of his local IEEE chapter and FIRST robotics organizations.

Aaron Burry is a principal scientist at Conduent, where his work focuses on enabling business process solutions using computer vision. His personal research interests include robust methods for object localization and tracking from video data and adaptive computer vision approaches that enable highly scalable/deployable solutions. He received both his bachelor's and his master's degrees in electrical engineering from the RIT.

Robert Loce is a patent technical specialist at Datto Inc, focusing on protecting business data and computer disaster recovery. He is a former research fellow at PARC a Xerox Company leading projects aimed at public safety. He has a PhD in imaging science (RIT), holds over 240 US patents in imaging systems, recently completed editing/coauthoring a book entitled Computer Vision and Imaging in Intelligent Transportation Systems, and is an SPIE fellow and IEEE senior member.