VIOLA: Video Labeling Application for Security Domains

Elizabeth Bondi[1], Fei Fang[2], Debarun Kar[1], Venil Noronha[1], Donnabell Dmello[1], Milind Tambe[1], Arvind Iyer[3], and Robert Hannaford[3]

[1]University of Southern California, Los Angeles, CA,
[2]Carnegie Mellon University, Pittsburgh, PA,
[3]AirShepherd, Berkeley Springs, WV

Abstract. Advances in computational game theory have led to several successfully deployed applications in security domains. These game-theoretic approaches and security applications learn game payoff values or adversary behaviors from annotated input data provided by domain experts and practitioners in the field, or collected through experiments with human subjects. Beyond these traditional methods, unmanned aerial vehicles (UAVs) have become an important surveillance tool used in security domains to collect the required annotated data. However, collecting annotated data from videos taken by UAVs efficiently, and using these data to build datasets that can be used for learning payoffs or adversary behaviors in game-theoretic approaches and security applications, is an under-explored research question. This paper presents VIOLA, a novel labeling application that includes (i) a workload distribution framework to efficiently gather human labels from videos in a secured manner; (ii) a software interface with features designed for labeling videos taken by UAVs in the domain of wildlife security. We also present the evolution of VIOLA and analyze how the changes made in the development process relate to the efficiency of labeling, including when seemingly obvious improvements surprisingly did not lead to increased efficiency. VIOLA enables collecting massive amounts of data with detailed information from challenging security videos such as those collected aboard UAVs for wildlife security. VIOLA will lead to the development of a new generation of game-theoretic approaches for security domains, including approaches that integrate deep learning and game theory for real-time detection and response.

Keywords: UAV, security, video surveillance, labeling application

1 Introduction

Security has already widely benefited from the use of game theory to develop better protection strategies. Game-theoretic approaches have led to applications that have been successfully deployed in infrastructure security domains such as protecting airports, ports and metro systems [28], as well as in green security domains such as protecting wildlife, forests, and fisheries [8, 11, 9]. In these

game-theoretic approaches and security applications, input data are needed to determine the payoff structure of the game, to learn the behavioral models of the players, and to predict where attackers are more likely to attack. In previous efforts, the data were provided by domain experts directly [24], recorded by practitioners in the field over months or years [19, 13], or collected through human subject experiments on platforms such as Amazon Mechanical Turk (AMT) [12].

With the recent use of unmanned aerial vehicle (UAV) technology in security domains, videos taken by UAVs have become an emerging source of massive data [10], especially in the domain of wildlife protection (e.g., the PAWS security games application [8]). For example, detecting wildlife from UAV videos can help estimate the animal distribution density, which decides the payoff structure of the game. Detecting poachers and their movement patterns could lead to successful learning of attackers' behavioral models, which is an important topic in security games [20, 12]. Data collected from UAVs can not only be used to provide input data to the game-theoretic models, but can also enable the development of a new generation of game-theoretic tools for security. The data can be used to train or fine-tune a deep neural network to automatically detect attackers from the video taken by the UAVs in real-time.

Unfortunately, collecting labeled data from videos taken by UAVs can be a labor-intensive, time-consuming task. To our knowledge, there is no existing application that focuses on assisting in the labeling of videos taken by UAVs in security domains. Existing applications for labeling images [6, 7] cannot be directly applied to labeling videos, as treating each frame as a separate image can lead to inefficiency since it does not exploit the correlation between frames. Video labeling applications such as VATIC [29] attempt to choose key frames for labeling, or track objects through the video. However, in UAV videos with camera motion, possibly collected using a different wavelength, these methods may not apply and may lead to inaccurate results or extra work for labelers, since the position of the objects in the video may change abruptly and the lack of color bands makes the tracking much more difficult. Furthermore, these applications are often paired with AMT to get labeled video datasets from online workers. However, in a security domain with sensitive data, meaning data that would provide attackers with some knowledge of defenders' strategies should it be shared, it may be undesirable to use AMT. This would then require finding labelers, and setting up an internal system to keep the process organized.

In this paper, we focus on better collection of labeled data from UAVs to provide input for game-theoretic approaches for security, and in particular to security game applications for wildlife conservation such as PAWS [8]. There has been work on labeling tools in domains such as computer vision and cyber security [6, 5], but there exists no work on labeling tools for game-theoretic approaches in security domains. Most previous work on game theory for security ignores where the payoffs and behavioral models come from, and we fill the gap.

In particular, we will focus on labeling videos taken by long wave thermal infrared (hereafter referred to as thermal infrared) cameras installed on UAVs, in the domain of wildlife security. We present VIOLA (VIdeO Labeling Applica-

tion), a novel application that assists labeling objects of interest such as wildlife and poachers. VIOLA includes a workload distribution framework to efficiently gather human labels from videos in a secured manner. We distribute the work of labeling the videos and reviewing the labels amongst a small group of labelers to ensure efficiency and data security. VIOLA also provides an easy-to-use interface, with a set of features designed for UAV videos in the wildlife security domain, such as allowing for moving multiple bounding boxes simultaneously and tracking bright spots in the video automatically. We will also discuss the various stages of development to create VIOLA, and we will analyze the impact of different labeling procedures and versions of the labeling application on efficiency, with a particular emphasis on the surprising results that showed some changes did not increase the efficiency.

2 Related Work

Game-theoretic approaches have been widely used in infrastructure and green security domains [28]. In green security domains such as protecting wildlife from poaching, multiple research efforts in artificial intelligence and conservation biology have attempted to estimate wildlife distribution and poacher activities [8]; such efforts often rely on months or years of recorded data [19, 13]. With the recent advances in unmanned aerial vehicle (UAV) technology, there is an opportunity to provide detailed data about wildlife and poachers for game-theoretic approaches. Since a poacher is rewarded for successfully poaching wildlife, the wildlife distribution determines the payoff structure of the game. Poachers' behavioral models can be inferred from poaching activities and be used to design better patrol strategies with game-theoretic reasoning. In addition, game-theoretic patrolling with alarm systems [1, 4] has been studied. UAVs can provide input for such systems in real-time using computer vision, particularly by detecting attackers or suspicious human beings in the UAV videos.

Detecting attackers in the UAV videos is related to object detection. Recently, great progress has been achieved in computer vision by deep learning in object detection and recognition [26, 25]. However, state-of-the-art detectors cannot be directly applied to our aerial videos because most methods focus on detection in high resolution, visible spectrum images. An alternative approach to this detection is to track moving objects throughout videos. Tracking of both single and multiple objects in videos has been studied extensively [31]. These methods also rely on high resolution visible spectrum videos. Single object trackers use discriminant features from high resolution videos to establish correspondences [14]. Much of multi-object tracking research is directed towards pedestrians [3, 32, 17], and primarily focuses on visible spectrum videos with high resolution, or videos taken from a fixed camera (except [17]).

Simpler and more general tracking algorithms exist that do not necessarily have these dependencies, such as the Lucas-Kanade tracker for optical flow [15], popular in the OpenCV package, and general correlation-based tracking [16]. Small moving objects can also be detected by a background subtraction method

after applying video stabilization [22]. Because these methods are more general, they are still applicable to our domain and were explicitly tested, but still did not perform well in many cases. For example, since the video stabilization and background subtraction method assumes a planar surface, in the case of more complex terrain, there were many noisy detections. Instead of using tracking for detection, we therefore decided to focus on deep learning.

In order to use deep learning-based detection methods with aerial, thermal infrared data, hand-labeled training data are required to fine-tune the networks or even train them from scratch. In addition to video labeling applications such as VATIC [29], there has been work on semi-automatic labeling [30] and label propagation [2] which combines the effort of human labelers and algorithms to speed up the labeling process for videos. This work often focuses on how to select the frames for human labelers to label and how to propagate the labels for the remaining frames. This is difficult for our domain because of the motion of UAVs, and because it is often hard for humans to tell which objects are of interest without seeing the object's motion. As a result, we sought to develop our own labeling application, VIOLA. The first key component of the application is a workload distribution framework. A common framework for image and video labeling is a majority voting framework [18, 23, 21, 27]. VIOLA uses a framework based upon [7] to efficiently gather labels from a small group of labelers. We examine the framework further in Sec. 6 and Sec. 7.

3 Domain

There has recently been increased use of UAVs for security surveillance. UAVs are able to cover more ground than a stationary camera and can provide the defenders more advanced notice of a potential threat. To detect suspicious human activities at night, the UAVs can be equipped with thermal infrared cameras. This is the type of UAV video we deal with in our domain, since poaching often occurs at night. We will specifically be able to use these types of data to detect poachers and provide advanced notice to park rangers, and use these detections to provide input for patrol generation tools such as PAWS.

In order to accomplish this, we need labeled data from the thermal infrared, UAV videos in the form of rectangular "bounding boxes" for objects of interest (animals and poachers) in each frame, with a color corresponding to their classification. However, the movement of UAVs and the thermal infrared images make it extremely difficult to label videos in this domain. First, thermal infrared cameras are low-resolution, and typically show warmer objects as brighter pixels in the image, although the polarity could be reversed occasionally. Different phenomena could also cause brighter pixels without a warm object. For example, the ground warms during the day, and then emits heat at night, which can be reflected under a tree canopy and lead to an amplified signal that might look like a human or animal. Furthermore, vegetation often looks bright and similar to objects of interest, as in Fig. 1, where there are three humans labeled with bounding boxes, amongst many other bright objects. Second, since the data are

captured aboard a moving UAV, these data often vary drastically. For example, the resolution, and therefore size of targets, is very different throughout our dataset because the UAV flies at varying altitudes.

In addition to difficult, variable video data to begin with, some videos may have many objects of interest in them, whereas some videos may not have any objects of interest at all. It sometimes takes a long time to determine if there are any objects of interest, and it also often takes a long time to label when there are many objects of interest. To illustrate the variation in the number of objects of interest, we analyze the historical videos we get from our collaborator. Fig. 2 shows a histogram of the average number of labels per frame, meaning that all frames in the video were counted, regardless of whether or not they were labeled, and a histogram of the average number of labels per labeled frame, meaning only frames that had at least one label were counted.

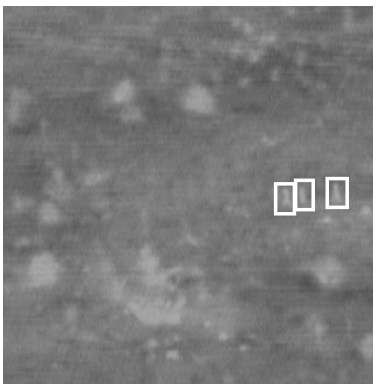


Fig. 1: An example of a thermal infrared frame, where the three humans outlined by the white boxes look very similar to the surrounding vegetation.

Although we focus on UAV videos in wildlife security domains, similar challenges in UAV videos in other security domains can be expected. Therefore, the application VIOLA we introduce in this paper can potentially be applied to other security domains to provide input for game-theoretic approaches.

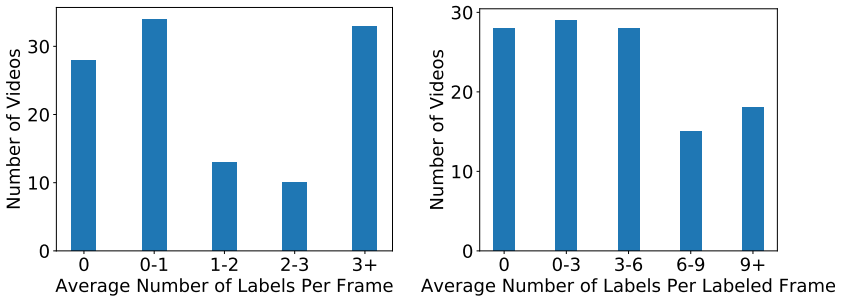


Fig. 2: A histogram with the number of videos for average objects of interest per frame (left), and the average objects of interest per labeled frame (right).

4 Example Game-Theoretic Uses

We now provide two more specific examples of game-theoretic approaches that may be derived from the data acquired using VIOLA. First, we focus on using the labeled data directly for behavioral models. Second, we discuss using the labeled data to train deep learning models for further data analysis.

With the labels provided by VIOLA and information about each frame, such as GPS and camera angle, we can locate poachers exactly throughout labeled videos. As such, we know the exact location of poaching activities and could use this information to learn how the poachers make decisions on where to poach. In particular, we could use an existing behavioral model, such as SUQR [20], and the location of poaching activity derived from the labels to update or improve the behavioral model for poachers, which would better inform patrol strategies. Furthermore, we could analyze the movement of the poachers, and a new behavioral model could be built using these movement patterns, in which poachers could choose a path instead of simply choosing a target to attack. This new behavioral model could be exploited to plan game-theoretic patrols.

In addition to directly using the labels from VIOLA for behavioral models, the labels could be used to train a deep learning model to automatically identify poachers in real-time video streams. Similarly, we could use the output from the deep learning algorithm for behavioral models, and the automated identification would allow us to circumvent the need for human labelers when incorporating data collected in the future into the behavioral models. Moreover, patrollers could make online decisions during patrols without the need for additional personnel to monitor the videos in the field. The ability to make online decisions during patrols could lead to new models of game-theoretic patrolling. Patrols could even be made for the UAVs themselves, which could introduce some behavioral challenges. The UAVs could also potentially be used as a deterrent, so flying UAVs could serve to both detect and deter poaching activities, while also collecting more data. In short, VIOLA has the potential to provide data that will better inform behavioral models and patrollers in the field, and introduce new questions that can be answered using game-theoretic approaches.

5 VIOLA

The main contribution of this paper is VIOLA, an application we developed for labeling UAV videos in wildlife security domains. VIOLA includes an easy-to-use interface for labelers and a basic framework to enable efficient usage of the application. In this section, we first discuss the user interface and then the framework for work distribution and training process for labelers.

5.1 User Interface of VIOLA

The user interface of VIOLA was written in Java and Javascript, and hosted on a server through a cloud computing service so it could be accessed using a URL from anywhere with an internet connection.

Before labeling, labelers were asked to login to ensure data security (Fig. 3a). The first menu that appears after login (Fig. 3b) asks the labeler which mode they would like, whether they would like to label a new video or review a previous submission. Then, after choosing "Label", the second menu (Fig. 3c) asks them to choose a video to label. Fig. 4 is an example of the next screen used for labeling, also with sample bounding boxes that might be drawn at this stage. Along the top of the screen is an indication of the mode and the current video name, and along the bottom of the screen is a toolbar. First, in the bottom left corner, is a percentage indicating progress through the video. Then, there are four buttons used to navigate through the video. The two arrows move backwards or forwards, the play button advances frames at a rate of one frame per second, and the square stop button returns to the first frame of the video. The next button is the undo button, which removes the bounding boxes just drawn in the current frame, just in case they are too tiny to easily delete. Also to help with the nuisance of creating tiny boxes by accident while drawing a new bounding box or while moving existing bounding boxes, there is a filter on bounding box size. The trash can button deletes the labeler's progress and takes them back to the first menu after login (Fig. 3b). Otherwise, work is automatically saved after each change and re-loaded each time the browser is closed and re-opened. The application asks for confirmation before deleting the labeler's progress and undoing bounding boxes to prevent accidental loss of work. The check-mark button is used to submit the labeler's work, and is only pressed when the whole video is finished. Again, there is a confirmation screen to avoid accidentally submitting half of a video. The copy button and the slider will be described further in Sec. 6. The eye button allows the labeler to toggle the display of the bounding boxes on the frame, which is often helpful during review to check that the labels are correct. Finally, the question mark button provides a help menu with a similar summary of the controls of the application (Fig. 5). Notice the bounding boxes surrounding the animals in this video are colored red. Humans would be colored blue. This is also included in the help menu.

To draw bounding boxes, the labeler can simply click and drag a box around the object of interest, then click the box until the color reflects the class. Deleting a bounding box is done by pressing SHIFT and click, and selecting multiple bounding boxes is done by pressing CTRL and click, which allows the labeler to move multiple bounding boxes at once. Finally, while advancing frames, bounding boxes drawn in the current frame are moved to the next frame. It only happens the first time a frame is viewed since it could otherwise add redundant bounding boxes or replace the bounding boxes originally added by the labeler.

If "Review" is chosen in the first menu after login, the second menu also asks the labeler to choose a video to review, and then a third menu (Fig. 3d) asks them to choose a labeling submission to review. It finally displays the video with the labels from that particular submission, and they may begin reviewing the submission. The two differences between the labeling and review modes in the application are (i) that the review mode displays an existing set of labels and (ii) that labels are not moved to the next frame in review mode.

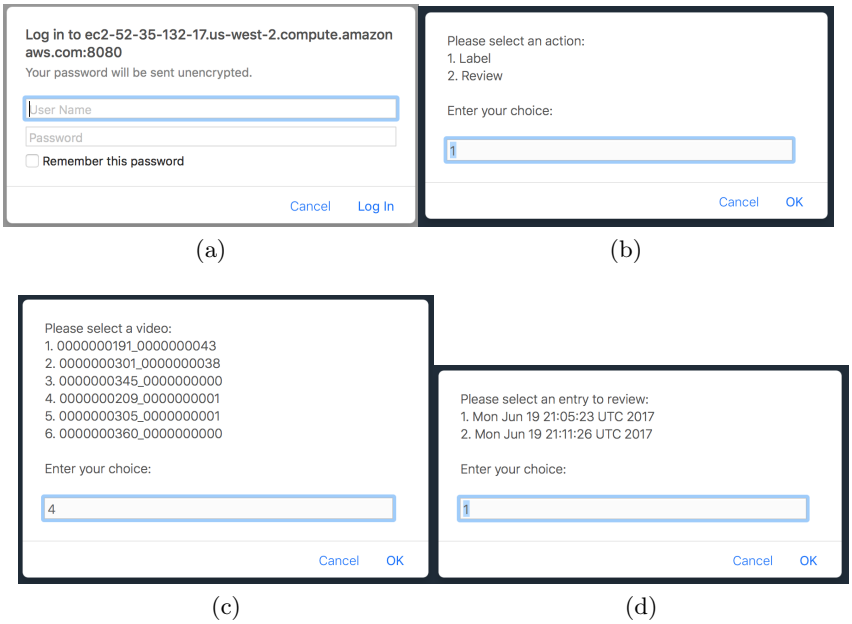


Fig. 3: The menus to begin labeling.

5.2 Use of VIOLA

Our goal in labeling the challenging videos in the wildlife security domain is first to keep the data secure, and second, to collect more usable labels to provide input for game-theoretic tools for security. In addition, we aim for getting exhaustive labels with high accuracy and consistency. To achieve these goals, we distribute the work among a small group of labelers in a secured manner, assign labelers to either provide or review others' labels, and supply guidelines for the labelers.

Distribution of Work To keep the data (historical videos from our collaborators) secure, instead of using AMT, we recruit a small group of labelers, in this work 13. Labelers are given a username and password to access the labeling interface, and the images on the labeling interface cannot be downloaded.

In order to achieve label accuracy, we use a framework of label and review. The idea is simply that one person labels a video, and another person checks, or reviews, the labels of the first person. By checking the work of the labeler, the reviewer must agree or disagree with the original set of labels instead of creating their own. Upon disagreement, the reviewer can change the original labels. This was primarily chosen because it was clean, leading to one set of final labels.

We use spreadsheets to share both assignments and completion progress with the team of labelers. We ask labelers to include the time it took for them to complete their assignment in order to help make future assignments more reasonable in terms of time commitment, and in order to track the efficiency and success of

Fig. 4: An example of a frame (left) and labeled frame (right) in a video. This is the next screen displayed after all of the menus, and allows the labeler to navigate through the video and manipulate or draw bounding boxes throughout.

the application itself. In addition, we split long videos into segments to make it easier to respect labelers' time commitments, and to finish extremely long videos quickly. There are also some videos that have long periods of nothingness, which are easier to ignore when the video is split.

Guidelines and Training for Labelers In order to achieve accuracy and consistency of labels, we provide guidelines and training for the labelers. During the training, we show the labelers several examples of the videos and point out the features of interest. We provide them with general guidelines on how to start labeling a video, as below.

In general, the process for labeling should be:

- Watch the video once all the way through and try to decide what you see.
- Once you have an idea of what is happening in the video by going through it, return to the beginning of the video and start labeling.
- Make and move bounding boxes.
- Send screenshots (including the percentage in the videos) if you need help.

In general, the process for reviewing should be:

- Refer to the guidelines and special circumstances directions.
- Go through the video, and use the eye button to check the original labels.
- Move, create, or delete bounding boxes as necessary, either as you go or after watching the whole video. Try not to resize the bounding boxes unless they are much too big or too small. Only change the classification and add or delete boxes if certain, and please confirm with us if not.
- Send screenshots (including the percentage in the videos) if you need help.

We also provide special instructions for the videos in our domain of interest, including a few key clues. For example, animals tend to be in herds, obviously shaped like animals, and/or significantly brighter than the rest of the scene, and humans tend to be moving. We also provide the following additional guidelines.

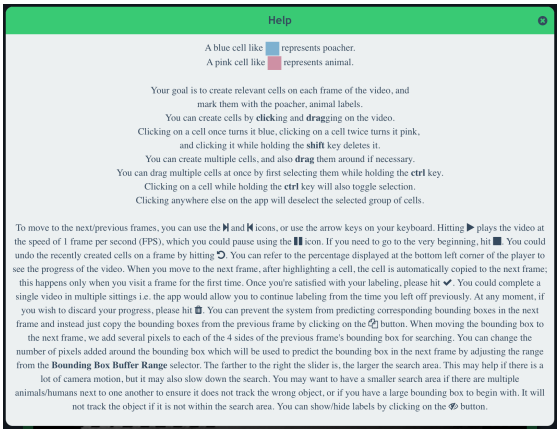


Fig. 5: Help screen detailing the controls of the application (? icon).

Fig. 6: Three consecutive frames where the middle frame has ghosting. The middle frame is "in between" the left and right frames.

Directions for special circumstances:

- Only label when objects are bright since the polarity changes occasionally
- If something is occluded completely: do not label
- If something is occluded but you can still see most features of them: label
- If something is shaped like a human but never moves: do not label
- If something is cutoff halfway in/out of the frame: do not label
- If there are "ghosts" (Fig. 6): do not label
- If you cannot recognize an individual (i.e., distinct poachers and animals): do not label

The final instruction about distinct objects is one of the more difficult instructions to follow in practice because often, the aerial view and small targets make it difficult to tell if there are one or more animals. The movement instruction is also difficult, since with so few pixels on objects plus camera motion, it sometimes looks like objects are moving that are not. In these ambiguous cases, labelers are encouraged to seek help. In cases of disagreement after discussion, we err on the side of caution and only label certain objects.

Table 1: Changes made throughout development.

Version	Change	Date of Change	Brief Description
1	-	-	Draws and edits boxes, navigates video, copies boxes to next frame
2	Multiple Box Selection	3/23/17	Moves multiple boxes at once, to increase labeling speed
3	Five Majority to Review	3/24/17	Requires only two people per video instead of five to improve overall efficiency
4	Labeling Days	4/12/17	Has labelers assemble to discuss difficult videos
5	Tracking	6/17/17	Copies and automatically moves boxes to next frame

6 Development

Thanks in large part to feedback provided by the labelers, we were able to make improvements throughout the development of the application to the current version discussed in Sec. 5.1. In the initial version of the application, we had five people label a single video, and then automatically checked for a majority consensus among these five sets of labels. We used the Intersection over Union (IoU) metric to check for overlap with a threshold of 0.5 [7]. If at least three out of five sets of labels overlapped, it was deemed to be consensus, and we took the bounding box coordinates of the first labeler. Our main motivation for having five opinions per video was to compensate for the difficulty of labeling thermal infrared data, though we also took into account the work of [18] and [23]. The interface of the initial version allowed the user to draw and manipulate bounding boxes, navigate through the video, save work automatically, and submit the completed video. Boxes were copied to the next frame and could be moved individually. To get where we are today, the changes were as listed in Table 1.

The most significant change made during the development process was the transition from five labelers labeling the same video and using majority voting to get the final labels (referred to as "MajVote") to having one labeler label the video followed by a reviewer reviewing the labels (referred to as "LabelReview"). We realized that having five people label a single video was very time consuming, and the quality of the labels was still not perfect because of the ambiguity of labeling thermal infrared data, which led to little consensus. Furthermore, when there was consensus, there were three to five different sets of coordinates to consider. Switching to LabelReview eliminated this problem, providing a cleaner and also time-saving solution. Another change, "Labeling Days", consisted of meeting together in one place for several hours per week so labelers were able to discuss ambiguities with us or their peers during labeling. Finally, the tracking algorithm (Alg. 1) was added to automatically track the bounding boxes when the labeler moves to the new frame. The goal was to improve labeling efficiency,

Algorithm 1 Basic Tracking Algorithm

```
1: bufferPixels ← userInput
2: for all boxesPreviousFrame do
3:     if boxSize > sizeThreshold then
4:         newBoxCoordinates ← boxCoordinates
5:     else
6:         searchArea ← newFrame[boxCoordinates + bufferPixels]
7:         thresholdedImage ← Threshold(searchArea, threshold)
8:         components ← ConnectedComponents(thresholdedImage)
9:         if numberComponents > 0 then
10:            newBoxCoordinates ← GetLargestComponent(components)
11:        else
12:            newBoxCoordinates ← boxCoordinates
13:        end if
14:    end if
15:    CopyAndMoveBox(newFrame, newBoxCoordinates)
16: end for
```

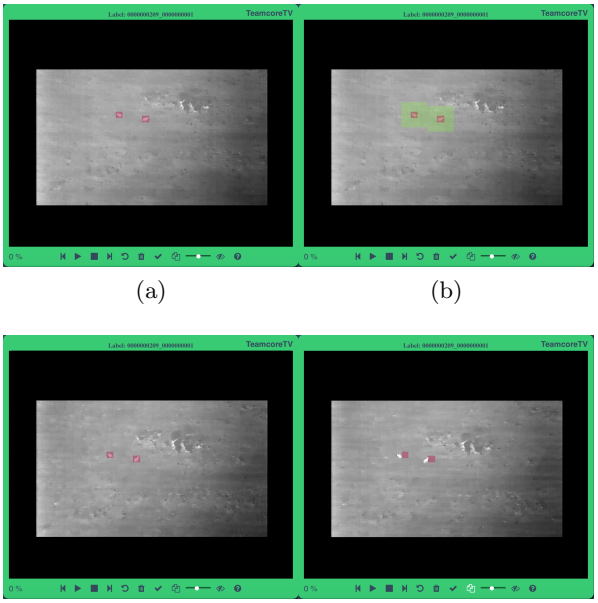
as the labelers would be able to label a single frame, then simply check that the labels were correct.

An example of the tracking process in use is shown in Fig. 7. First, the labeler drew two bounding boxes around the animals (Fig. 7a), then adjusted the search size for the tracking algorithm using the slider in the toolbar (Fig. 7b). The tracking algorithm was applied to produce the new bounding box location (Fig. 7c). In contrast, the copy feature, activated when the copy button was selected on the toolbar, only copied the boxes to the same location (Fig. 7d). In this case, since there was movement, and the animals were large and far from one another, the tracking algorithm correctly identified the animals in consecutive frames. If several bright objects were in the search region, it could track incorrectly and copying could be better. One direction of future work is to improve the tracking algorithm by setting thresholds automatically and accounting for close objects.

7 Analysis

In this section, we analyze how the changes we made during the development of VIOLA affect labeling efficiency. To do this, we examine two questions: (i) how the changes affect the overall efficiency of the data collection process, which is measured by the total person time needed to get a final label – a label confirmed by the five majority voting or the reviewer that can be used for game-theoretic analysis or deep learning algorithms; (ii) how the changes affect the individual efficiency, or the person time needed for an individual labeler or reviewer to provide or check a label. In addition, we examine whether other desired properties of the data collection process, such as exhaustiveness, have been achieved.

To analyze efficiency, we first went through the person time data collected during VIOLA's development. Any changes made were deployed immediately



(a) (b)

(c) (d)

Fig. 7: A sample labeling process.

to make faster progress. These person time data came from different videos and labelers. They inherently took different amounts of time to label, since the videos varied in their content. To mitigate the intrinsic heterogeneity, we divide the videos into four groups, $(0, 1)$, $[1, 2)$, $[2, 3)$, and $[3, +\infty)$, based on the average number of labels per frame, since it was an important indicator of the difficulty of labeling a video. There were other factors affecting the difficulty of labeling videos, so videos in the same group may still have had high variation. Because of this, we remove the top and bottom 5% of time per label entries.

Also due to these concerns, we collected additional person time data in a more controlled environment. We gave six unique videos that contained animals but no poachers to the labelers to label. The labelers had not seen these videos previously. We distributed the work among the labelers so as to get one set of final labels for each video under each of the versions of VIOLA (as shown in Table 1). We asked the labelers to label for no more than 15 minutes on each video. To accommodate the labelers' schedules and coordinate their schedules to set up meetings, which are necessary for LabelDays and Tracking, we gave the labelers 2 to 4 days to label the videos under each version. As such, it was difficult to get multiple sets of labels for each video or get labels for more videos. Some labelers were not able to complete checking all of the frames in the video within 15 minutes, so we use the minimum checked frame among labelers for each video under each version, and analyze efficiency using person time data up

Table 2: Versions tested in the additional tests.

Version Number	1	2	3	4	5
Version Name	Basic	MultiBox	Review	LabelDays	Tracking
Framework Used	MajVote	MajVote	LabelReview	LabelReview	LabelReview
Test Order	Fourth	Third	First	Second	Fifth

until that frame only. Also, note that since some labelers were asked to label the same video multiple times under different versions, the labelers likely got faster as time went on. To mitigate these effects, we randomly ordered the five versions of VIOLA for them to label. The order is shown in Table 2.

We will proceed in this section by first focusing on the impact of the key change in the labeling framework from MajVote to LabelReview on the overall efficiency. We will then check each version of VIOLA to understand the impact of other changes. Because of the surprising results, we will particularly examine videos in which these features helped and in which they did not.

7.1 From MajVote to LabelReview

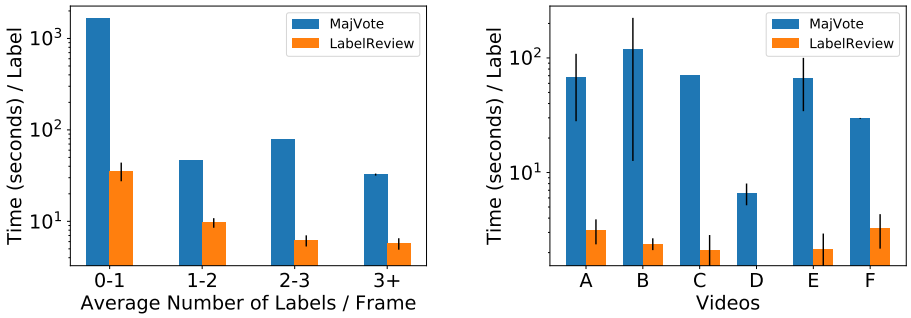
Fig. 8a and Fig. 8b show the comparison on overall efficiency between MajVote and LabelReview. The total person time per final label is lower on average when we use LabelReview, based on data collected through both the development process and additional tests. During the development process, there were only seven videos for which we got final labels from five full sets of labels using MajVote, two of which did not produce any consensus labels. There were more than 70 videos for which we got final labels through LabelReview. During the additional tests, we tested two versions using MajVote and three versions using LabelReview, which means the value of each bar is averaged over two or three samples, respectively. We exclude one sample for Video C where no consensus labels were achieved through MajVote. The LabelReview efficiency for Video D is 0.63 with a standard error of 0.09 but it is too small to appear in Fig. 8b.

In addition to having more labelers involved, one reason that MajVote leads to a higher person time per final label is the lack of consensus. Fig. 9 shows that there were large discrepancies in the number of labels between individual labelers, which led to fewer consensus labels (zero in Videos I and M).

Fig. 10 shows that MajVote leads to many fewer final labels than LabelReview for the videos in the additional tests. This indicates that using LabelReview can get us closer to the goal of exhaustively labeling all of the objects of interest when compared to MajVote.

7.2 Impact of Other Changes

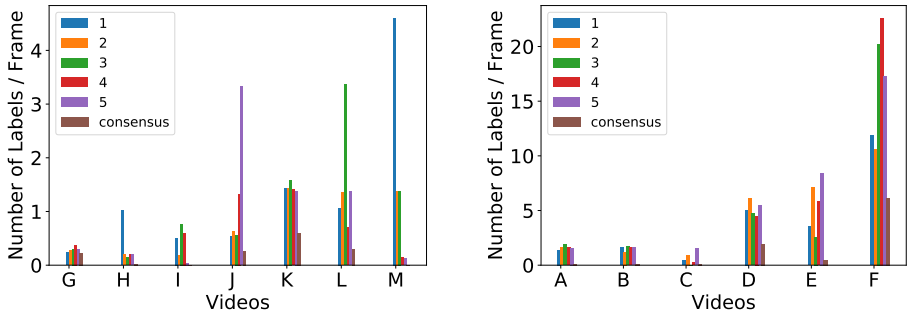
In this section, we examine the individual efficiency and overall efficiency of each version of VIOLA to analyze the impact of every other change we made during the development of VIOLA. For individual efficiency, we calculate person time



(a) Data from development process.

(b) Data from additional tests.

Fig. 8: Comparison of overall efficiency with different labeling frameworks.



(a) For the seven videos with five sets of labels during development process.

(b) For the six videos used in the additional tests under version Basic.

Fig. 9: Number of labels per frame for individual labelers and for consensus.

spent per label for each individual labeler or reviewer, regardless of whether that label has been confirmed to be a final label.

We first show results of individual efficiency based on person time data collected during the development process in Fig. 11. Person times per label for each video submission are colored to represent the group which is decided by the average number of labels per frame. Video submissions are reported by submission date since the date submitted indicates which version of the application was used for the video. The dates on which features were added, given in Table 1, are used to color the background of the plot. Finally, each submission is considered separately, to examine labeling or review efficiency only. Fig. 11 shows the person time per label for videos with low average number of labels per frame $(0-1)$ is higher than others for both labeling and reviewing. Fig. 12 shows the

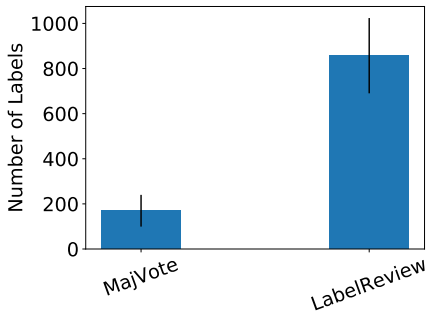


Fig. 10: Number of final labels for MajVote and LabelReview in additional tests.

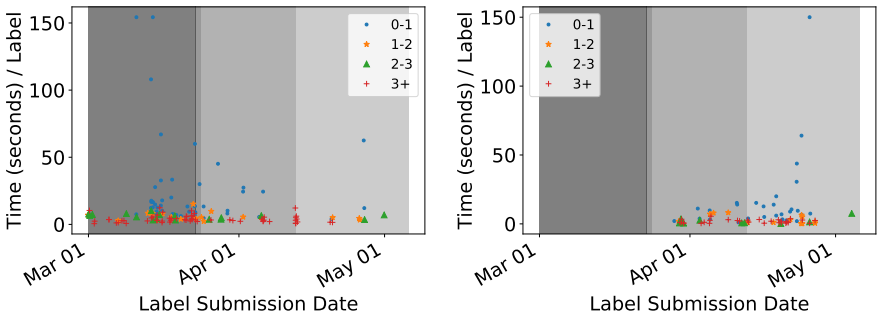


Fig. 11: Individual efficiency for each submission of labeling (left) and review (right) with data collected during the development process.

mean labeling and reviewing time per label within the timespan of each change during the development process.

We next examine the individual efficiency for labeling and reviewing in the additional tests (Fig. 13). The results of each test have been shown by video, since there were only five sets of labels in the tests with MajVote (Version 1-2) and only one set of labels in the tests with LabelReview (Version 3-5). The five sets of labels in the MajVote tests are averaged by video, and the standard error bars are included. Fig. 13 shows that each of the changes we made resulted in an improvement on the individual efficiency for some, but not all, of the videos.

Multiple Box Selection The feature of multiple box selection was added to improve the individual efficiency of labeling. Checking the first two groups in Fig. 12 and Fig. 13, we notice that surprisingly, this feature improves individual efficiency for some of the videos (e.g., Video F), but not all of the videos. One possible explanation is that in videos where there are many animals that do not move much over time, the changing position of the bounding boxes is mainly due to the movement of the camera. In this case, using multiple box selection and moving all of the bounding boxes in the same direction simultaneously is helpful.

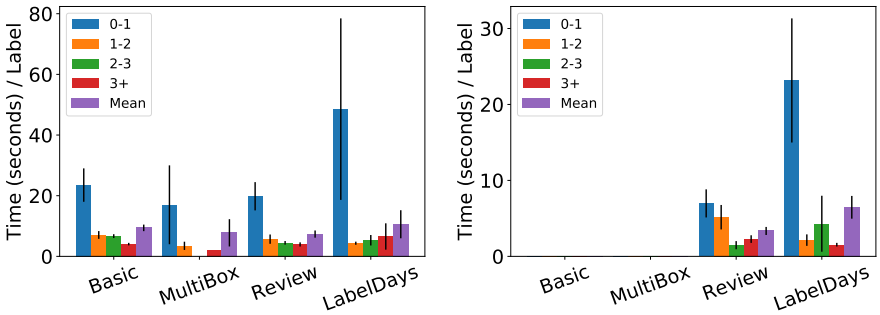


Fig. 12: Average individual efficiency of labeling (left) and review (right) with data collected during the development process.

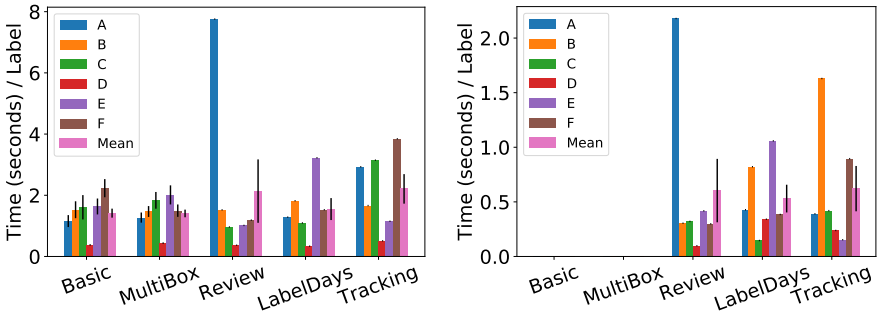


Fig. 13: Individual efficiency for each submission and average efficiency of labeling (left) and review (right) with data collected from the additional tests.

However, in other videos where there are only one or two animals in each frame, it may be faster to move the boxes separately, particularly if an animal moves.

Labeling Days Labeling days were introduced with the aim to increase the overall efficiency. Fig. 14 shows the average person time per final label has slightly reduced from Review to LabelDays during the additional tests, and the person time per final label has reduced for Videos A, C, and F. Fig. 14 also shows the number of final labels has remained the same on average. The results indicate that introducing labeling days may help improving the efficiency and exhaustiveness of labeling, at least for some more complex videos. Subjective feedback from the labelers also indicated that introducing labeling days made it easier for them to deal with ambiguous cases, when it is difficult to maintain consistency and accuracy despite the guidelines. However, Fig. 12, Fig. 13, and Fig. 14 show that introducing labeling days does not lead to an improvement on individual efficiency in all cases. It is possible that it increased the individual labeling time due to extra discussion, but it may have saved time during review. We plan to analyze the effects of labeling days in more detail in the future.

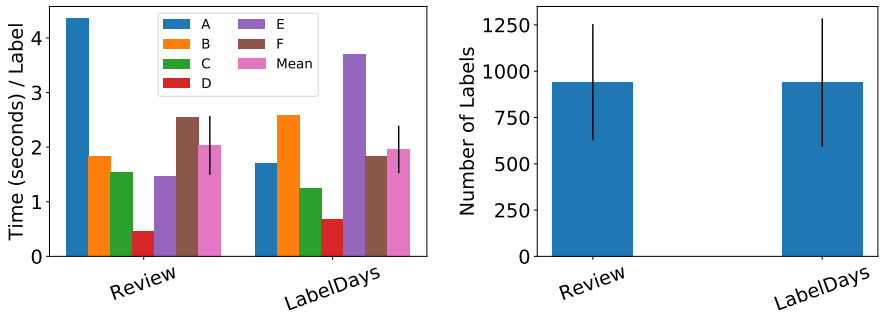


Fig. 14: Overall efficiency (left) and number of final labels (right) with version Review and LabelDays during the additional tests.

Tracking The tracking feature is the newest feature. We included it in the additional tests but it has not been deployed for the labelers to use. During the tests, we received positive feedback from labelers, particularly on videos in which animals were far apart and bright. In addition, the tracking feature was able to successfully track two animals in the first 10% of Video B, as shown in Fig. 7. Unexpectedly, the initial results from the additional tests do not show a positive effect on time per label or number of labels. We believe this is due to the fact that it does not find a brightness threshold automatically, and is likely to track the wrong object when multiple objects are within the same search region. We plan to continue developing this feature given its promise in the cases where animals are far apart and bright.

Summary This section thus shows that while some of our proposed improvements actually led to increased efficiency, particularly the switch from MajVote to LabelReview, in other cases (e.g., multiple box selection), surprisingly, it only increased efficiency in some videos. This result indicates that we must not simply add features on the intuition that they are bound to improve performance, as they may only be useful for certain videos.

8 Conclusions

In conclusion, we presented VIOLA, which provides a labeling and reviewing framework to gather labeled data from a small group of people in a secure manner, and a labeling interface with both general features for difficult video data, and specific features for our green security domain to track wildlife and poachers. We analyzed the impact of the framework and the features on labeling efficiency, and found that some changes did not improve efficiency in general, but worked only in particular types of videos.

We plan to utilize the labeled data we acquired in this work to estimate the animal distribution and predict poachers' movement patterns, which are important for game-theoretic approaches such as generating patrol strategies as in PAWS. In addition, we will use the dataset to train deep neural networks

to automatically detect wildlife and poachers in real-time, and develop novel game-theoretic approaches that incorporate real-time information to plan UAV and human patrol routes. VIOLA can be adopted to detect objects of interest in other types of surveillance videos, with widespread applications to various security domains.

9 Acknowledgments

This research was supported by UCAR N00173-16-2-C903, with the primary sponsor being the Naval Research Laboratory (Z17-19598). It was also partially supported by the Harvard Center for Research on Computation and Society Fellowship and the Viterbi School of Engineering Ph.D. Merit Top-Off Fellowship.

References

1. Alpcan, T., Basar, T.: A game theoretic approach to decision and analysis in network intrusion detection. In: Proceedings of 42nd IEEE Conference on Decision and Control. vol. 3, pp. 2595–2600. IEEE (2003)
2. Badrinarayanan, V., Galasso, F., Cipolla, R.: Label propagation in video sequences. In: CVPR. pp. 3265–3272. IEEE (2010)
3. Bae, S.H., Yoon, K.J.: Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning. In: CVPR (2014)
4. Basilico, N., Nittis, G.D., Gatti, N.: A security game combining patrolling and alarm–triggered responses under spatial and detection uncertainties. In: AAAI. pp. 397–403 (2016)
5. Catania, C.A., Bromberg, F., Garino, C.G.: An autonomous labeling approach to support vector machines algorithms for network traffic anomaly detection. Expert Systems with Applications 39(2), 1822–1829 (2012)
6. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR. pp. 248–255. IEEE (2009)
7. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. International Journal of Computer Vision 88(2), 303–338 (Jun 2010)
8. Fang, F., Nguyen, T.H., Pickles, R., Lam, W.Y., Clements, G.R., An, B., Singh, A., Tambe, M., Lemieux, A.: Deploying paws: Field optimization of the protection assistant for wildlife security. In: AAAI. pp. 3966–3973 (2016)
9. Haskell, W., Kar, D., Fang, F., Tambe, M., Cheung, S., Denicola, E.: Robust Protection of Fisheries with COmPASS. In: IAAI. pp. 2978–2983 (2014)
10. Hodgson, J.C., Baylis, S.M., Mott, R., Herrod, A., Clarke, R.H.: Precision wildlife monitoring using unmanned aerial vehicles. Scientific reports 6 (2016)
11. Johnson, M.P., Fang, F., Tambe, M.: Patrol Strategies to Maximize Pristine Forest Area. In: AAAI (2012)
12. Kar, D., Fang, F., Fave, F.D., Sintov, N., Tambe, M.: "A Game of Thrones": When Human Behavior Models Compete in Repeated Stackelberg Security Games. In: AAMAS (2015)
13. Kar, D., Ford, B., Gholami, S., Fang, F., Plumptre, A., Tambe, M., Driciru, M., Wanyama, F., Rwetsiba, A., Nsubaga, M., Mabonga, J.: Cloudy with a chance of poaching: Adversary behavior modeling and forecasting with real-world poaching data. In: AAMAS. pp. 159–167 (2017)

14. Kristan, M., Matas, J., Leonardis, A., Felsberg, M., Cehovin, L., Fernández, G., Vojir, T., Hager, G., Nebehay, G., Pflugfelder, R.: The visual object tracking vot2015 challenge results. In: ICCV Workshops. pp. 1–23 (2015)
15. Lucas, B.D., Kanade, T., et al.: An iterative image registration technique with an application to stereo vision (1981)
16. Ma, C., Yang, X., Zhang, C., Yang, M.H.: Long-term correlation tracking. In: CVPR. pp. 5388–5396 (2015)
17. Milan, A., Leal-Taixé, L., Reid, I., Roth, S., Schindler, K.: Mot16: A benchmark for multi-object tracking. arXiv preprint arXiv:1603.00831 (2016)
18. Nguyen, P., Kim, J., Miller, R.C.: Generating annotations for how-to videos using crowdsourcing. In: CHI '13 Extended Abstracts on Human Factors in Computing Systems. pp. 835–840 (2013)
19. Nguyen, T.H., Sinha, A., Gholami, S., Plumptre, A., Joppa, L., Tambe, M., Driciru, M., Wanyama, F., Rwetsiba, A., Critchlow, R., et al.: Capture: A new predictive anti-poaching tool for wildlife protection. In: AAMAS. pp. 767–775 (2016)
20. Nguyen, T.H., Yang, R., Azaria, A., Kraus, S., Tambe, M.: Analyzing the Effectiveness of Adversary Modeling in Security Games. In: AAAI. pp. 718–724 (2013)
21. Nguyen-Dinh, L.V., Waldburger, C., Roggen, D., Tröster, G.: Tagging human activities in video by crowdsourcing. In: ICMR. pp. 263–270 (2013)
22. Pai, C.H., Lin, Y.P., Medioni, G.G., Hamza, R.R.: Moving object detection on a runway prior to landing using an onboard infrared camera. In: CVPR. pp. 1–8. IEEE (2007)
23. Park, S., Mohammadi, G., Artstein, R., Morency, L.P.: Crowdsourcing micro-level multimedia annotations: The challenges of evaluation and interface. In: CrowdMM. pp. 29–34 (2012)
24. Pita, J., Jain, M., Western, C., Portway, C., Tambe, M., Ordonez, F., Kraus, S., Paruchuri, P.: Deployed ARMOR protection: The application of a game theroetic model for security at the Los Angeles International Airport. In: AAMAS (2008)
25. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: CVPR. pp. 779–788 (2016)
26. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: NIPS. pp. 91–99 (2015)
27. Sheng, V.S., Provost, F., Ipeirotis, P.G.: Get another label? improving data quality and data mining using multiple, noisy labelers. In: KDD. pp. 614–622 (2008)
28. Tambe, M.: Security and Game Theory: Algorithms, Deployed Systems, Lessons Learned. Cambridge University Press (2011)
29. Vondrick, C., Patterson, D., Ramanan, D.: Efficiently scaling up crowdsourced video annotation. International Journal of Computer Vision pp. 1–21
30. Yan, R., Yang, J., Hauptmann, A.: Automatically labeling video data using multi-class active learning. In: ICCV (2003)
31. Yilmaz, A., Javed, O., Shah, M.: Object tracking: A survey. Acm computing surveys (CSUR) 38(4), 13 (2006)
32. Zhang, L., Li, Y., Nevatia, R.: Global data association for multi-object tracking using network flows. In: CVPR. pp. 1–8. IEEE (2008)