Technical Report
1248

Augmented Annotation Phase 3

L. Lee

09 March 2020

Lincoln Laboratory
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LEXINGTON, MASSACHUSETTS

This material is based upon work supported under Air Force Contract No. FA8702-15-D-0001.

This report is the result of studies performed at Lincoln Laboratory, a federally funded research and development center operated by Massachusetts Institute of Technology. This material is based upon work supported under Air Force Contract No. FA8702-15-D-0001. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the U.S. Air Force.

Massachusetts Institute of Technology
Lincoln Laboratory

Augmented Annotation Phase 3

L. Lee
Group 46

Technical Report 1248

09 March 2020

Lexington Massachusetts

This page intentionally left blank.

EXECUTIVE SUMMARY

Automated visual object detection is an important capability in reducing the burden on human operators in many DoD applications. To train modern deep learning algorithms to recognize desired objects, the algorithms must be "fed" more than 1000 labeled images (for 55%–85% accuracy according to project Maven - Oct 2017 O6, Working Group slide 27) of each particular object. The task of labeling training data for use in machine learning algorithms is human intensive, requires special software, and takes a great deal of time. Estimates from ImageNet, a widely used and publicly available visual object detection dataset, indicate that humans generated four annotations per minute in the overall production of ImageNet annotations. DoD's need is to reduce direct object-by-object human labeling particularly in the video domain where data quantity can be significant. The Augmented Annotations System addresses this need by leveraging a small amount of human annotation effort to propagate human initiated annotations through video to build an initial labeled dataset for training an object detector, and utilizing an automated object detector in an iterative loop to assist humans in pre-annotating new datasets.

Augmented Annotation was divided into four phases:

1. Bootstrap Augmented Annotation: Quantify annotation efficiency of leveraging a small number of human initiated annotations to produce a large number of annotations using visual tracking. Sponsor supplied video data containing targets of interest with a requirement of 2000 annotations per target class. MIT Lincoln Laboratory (MIT LL) successfully completed this phase in three months and demonstrated performance improvements over a purely manual process of 20X to 25X. Annotations produced in Phase 1 are used in Phase 3 (current phase) to train an initial object detector.

2. Prototype Augmented Annotation bootstrap system. MIT LL completed this phase in six months and delivered the prototype in a visual machine.

3. Iterative Augmented Annotation process (current phase): Evaluate annotation generation efficiency improvement using the iterative process of the Augmented Annotation System compared to using only the bootstrapping process. Report on performance improvements and experiences or "lessons learned." Use results in design of a full Augmented Annotation System.

4. Develop and deploy full Augmented Annotation System, including both bootstrapping and iterative process, for user evaluation.

Augmented Annotation Phase 3 includes the following subtasks:

1. Prototype iterative training and testing process.

2. Prototype user interface to verify auto-generated annotation initializations.

3. Compare human effort level using both the bootstrapping process and with the addition of auto-generated annotation initializations.

4. Measure performance improvements with each additional iteration of semi-automated annotation initialization in both detector accuracy and reduction in human effort.

5. Report on learned experiences and quantify reduction in human effort.

6. Apply experiences in design of a full Augmented Annotation System.

MIT LL prototyped the iterative training process using YOLO-tiny as a default object detector with modifications to improve detection performance on small objects; built a prototype user interface for human verification/editing of automated pre-annotations; and measured performance improvements over using the bootstrapping process alone. Our experiments using the iterative process with automated pre-annotations showed an improvement of up to 3X over using the Augmented Annotation bootstrapping process alone, or up to 66X efficiency improvement when compared to a purely manual annotation effort.

We learned a number of lessons through our experiments using the iterative annotation/training stage of Augmented Annotation:

1. Effectiveness of AA3 iterative training/annotation cycle is dependent on detector performance.

2. The importance of data selection/balancing in the iterative training process. The iterative training process needs to train the object detector using data that is representative of future datasets. Training with a highly biased dataset could produce an object detector with significantly worse performance on future datasets and thus revert the Augmented Annotation process to bootstrapping stage.

3. Manual verification effort is a bottleneck to increasing annotation efficiency. Human attention is limited during the annotation effort to blocks of no more than 20 minutes.

4. User interface improvements are needed to maximize use of human effort in the annotation process.

The iterative process of Augmented Annotation, which includes training and applying a machine-learning-based object detector with human in the loop in producing annotations, has shown great potential in dramatically increasing the efficiency of human annotators by up to 66X. Based on the outcome of Augmented Annotation Phase 3, MIT LL recommends that sponsor proceed with a deployable implementation of the full Augmented Annotation System. We recommend that the deployable Augmented Annotation system include the following elements:

1. A bootstrapping component that leverages human initialized annotations and propagates them through video using visual tracking.

2. An interface for iterative training of a visual object detector, with potential to change the detector architecture.

3. A user interface allowing the user to verify automated pre-annotations generated by a visual object detector.

4. An automated training supervisor (active learner) that pre-processes new datasets and selects datasets for human annotators. The active learner takes the role of discovering datasets that improve the utilization of human effort and training data balancing to reduce the chance of a biased annotation dataset.

5. To reduce the burden on human annotators to verify all annotations, automated pre-annotations could be utilized directly in training through a framework such as Snorkel or CleanLab.

6. User interface improvements to decouple the time element of video from the annotation verification process.

This page intentionally left blank.

TABLE OF CONTENTS

This page intentionally left blank.

LIST OF ILLUSTRATIONS

This page intentionally left blank.

LIST OF TABLES

This page intentionally left blank.

1. BACKGROUND

Promising sources of information about potential adversaries or threat actors are video and imagery. Historically, this information is analyzed at great cost of time and effort by analysts, whom often lose peak performance within the first few minutes of viewing video. To take advantage of current and emerging machine learning, deep learning, and active learning techniques, a large amount of labeled training data is required. The “training data,” generally, must contain captured images (i.e., a picture or video) that contain each specified (target) object on which the training is focused and annotations that outline the targets of interest, either in bounding boxes or segmentations. This training must be performed for each object that the algorithm is attempting to detect. The resulting outcome of applying a trained object detector on images and videos should show a bounding box clearly outlining the object of interest. To train these algorithms to recognize desired objects, the algorithms must be “fed” more than 1000 labeled images (for 55%–85% accuracy according to project Maven - Oct 2017 O6, Working Group slide 27) of each particular object. For this reason and others, the generation of labeled training data at minimal cost (time and effort of the analysts) is essential for DoD to leverage advanced computing techniques in the future.

MIT Lincoln Laboratory developed the Augmented Annotation System which includes a bootstrapping process using human guided automation, and an iterative process to increase the quantity of annotated data (see figure below). The bootstrapping process, utilizing manual annotations to initialize automation (which applies visual tracking to extend the manual annotations) has been evaluated in a feasibility study (phase 1) funded by DoD and demonstrated a labeling efficiency increase 20 to 25 times higher than a manual process on sponsor supplied data.

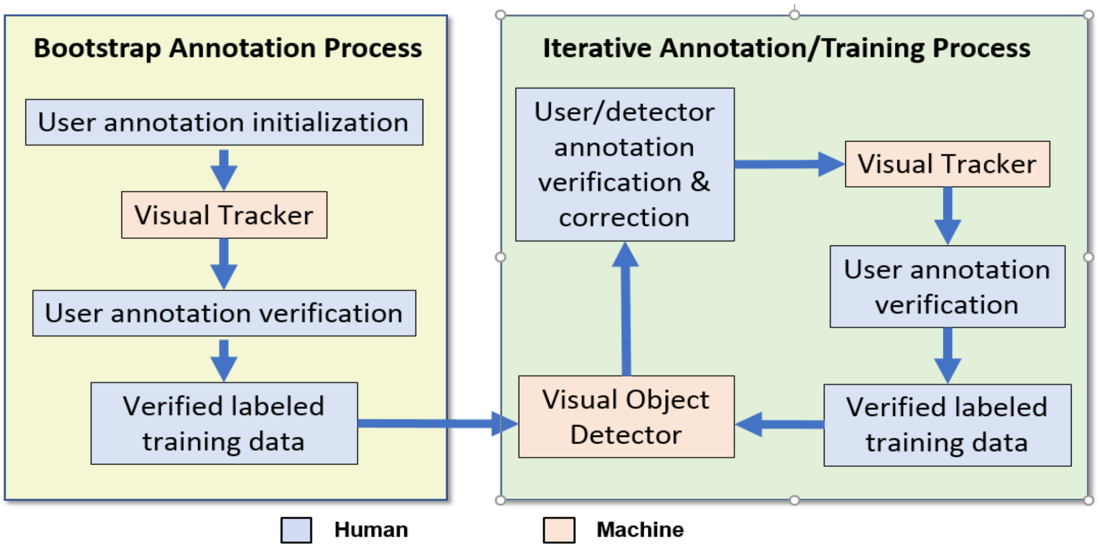

Bootstrap Annotation Process
User annotation initialization
Visual Tracker
User annotation verification
Verified labeled training data
Iterative Annotation/Training Process
User/detector annotation verification & correction
Visual Tracker
User annotation verification
Verified labeled training data
Visual Object Detector
Human
Machine

Figure 1. Augmented Annotation consists of a bootstrapping stage (yellow) and an iterative stage (green) that leverages computer vision/machine learning (red) and human expertise (blue).

Phase 2 of the Augmented Annotation program prototyped the bootstrapping stage for demonstration/evaluation by DoD users, which was demonstrated and delivered at the end of Phase 2.

Phase 3 (focus of this report) of the Augmented Annotation program is to evaluate the efficiency increase in the iterative process which utilizes an iterative training automated detector to initialize the visual tracker, which also has a human collaborating in the process to verify the automated detection initializations. This phase includes the following tasks:

1. Prototype iterative training and testing process

2. Prototype user interface to verify auto-generated annotation initializations

3. Compare human effort-level using both the bootstrapping process, and with the addition of auto-generated annotation initializations

4. Measure performance improvements with each additional iteration of semi-automated annotation initialization in both detector accuracy and reduction in human effort

5. Report on learned experiences and quantify reduction in human effort

6. Apply experiences in design of a full Augmented Annotation System

The rest of this report will discuss the Augmented Annotation Process and how it increases efficiency in an annotation effort (Section 2), the implementations MIT LL prototyped to evaluate the impact of an iterative training and annotation cycle (Section 3), the experiments to measure annotation efficiency improvements from iterative cycle (Section 4), the lessons learned from using the iterative cycle of the Augmented Annotation System (Section 5), and the design for a deployable Augmented Annotation System that integrates the bootstrapping process and the iterative process (Section 6).

2. AUGMENTED ANNOTATION

Visual object detectors require training data containing labelled visual data that appropriated describes the conditions under which the detector will be expected to function. To that end, a number of steps need to be applied in the effort to produce the training data. The first is the selection of datasets that appropriated reflect the conditions under which the detector must function. The second step of annotation effort is to locate the target classes of interest in among the datasets. For example, in video domain a human user may play a video collection to "seek" targets of interest. The amount of time needed for "seeking" is affected by complexity of scenes, the expertise of the user. In image domain this process takes the form of an annotator making a decision on each individual image whether a target class is contained in the image. The third step of annotation production is to label targets of interests by marking it with a bounding box or outline and assigning it with a target class label. The final step is to verify that the correctness of annotations with a subject matter expert. The verified annotations can be fully trusted for use in training visual object detectors.

The Augmented Annotation System addresses efficiency improvements in steps 2 through 4 of an annotation effort. The selection of an appropriate dataset is not directly addressed in Phase 3 of Augmented Annotation, but through our experiments we demonstrate the need for manually selecting appropriate datasets for a labeling effort. The visual tracking capability to propagate annotations across video reduces time and effort in marking once a target object is found. The application of a visual object detector to pre-annotate data with targets of interest reduces the annotator's time to "seek" targets of interest. Once the pre-annotations produced by a detector have been verified by a human and visually tracked, the final verification of tracked annotations operates at the level of tracks instead of individual annotations, which again reduces the burden on human annotators. Through our Augmented Annotation Phase 1 study, we demonstrated that visual tracking of user initialized annotations increases annotation efficiency by 20X. Through our current Phase 3 study, we are able to achieve a 3X improvement over visual tracking by pre-annotating using a visual object detector to reduce the "seek" time of finding targets of interest.

This page intentionally left blank.

3. IMPLEMENTATION

The Augmented Annotation System includes a user interface component which enables a user to perform the functions of initializing annotations, verification of pre-annotations, and verifications of tracked annotations. It also includes a backend computation component which contains capabilities for visual tracking and training/applying an object detector, and a backend data storage component which stores all the annotations and their status.

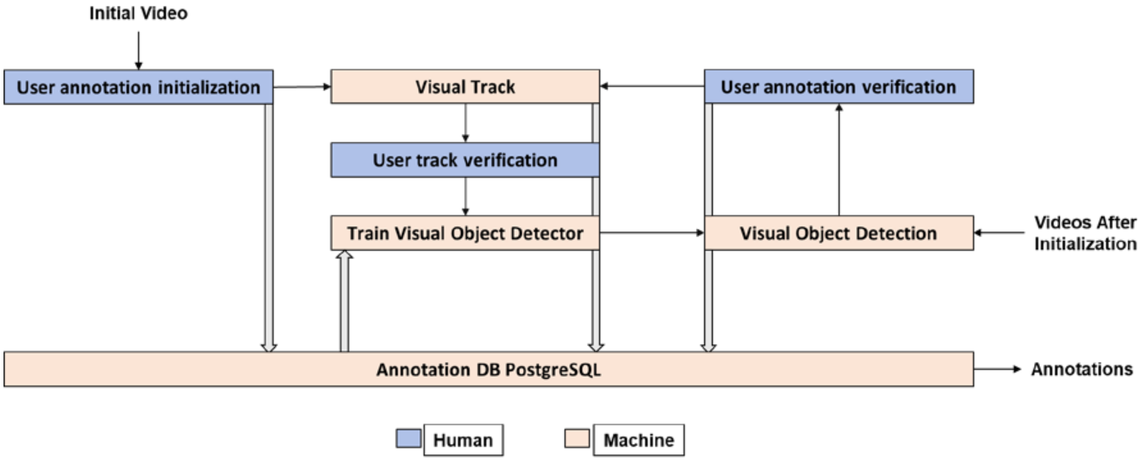


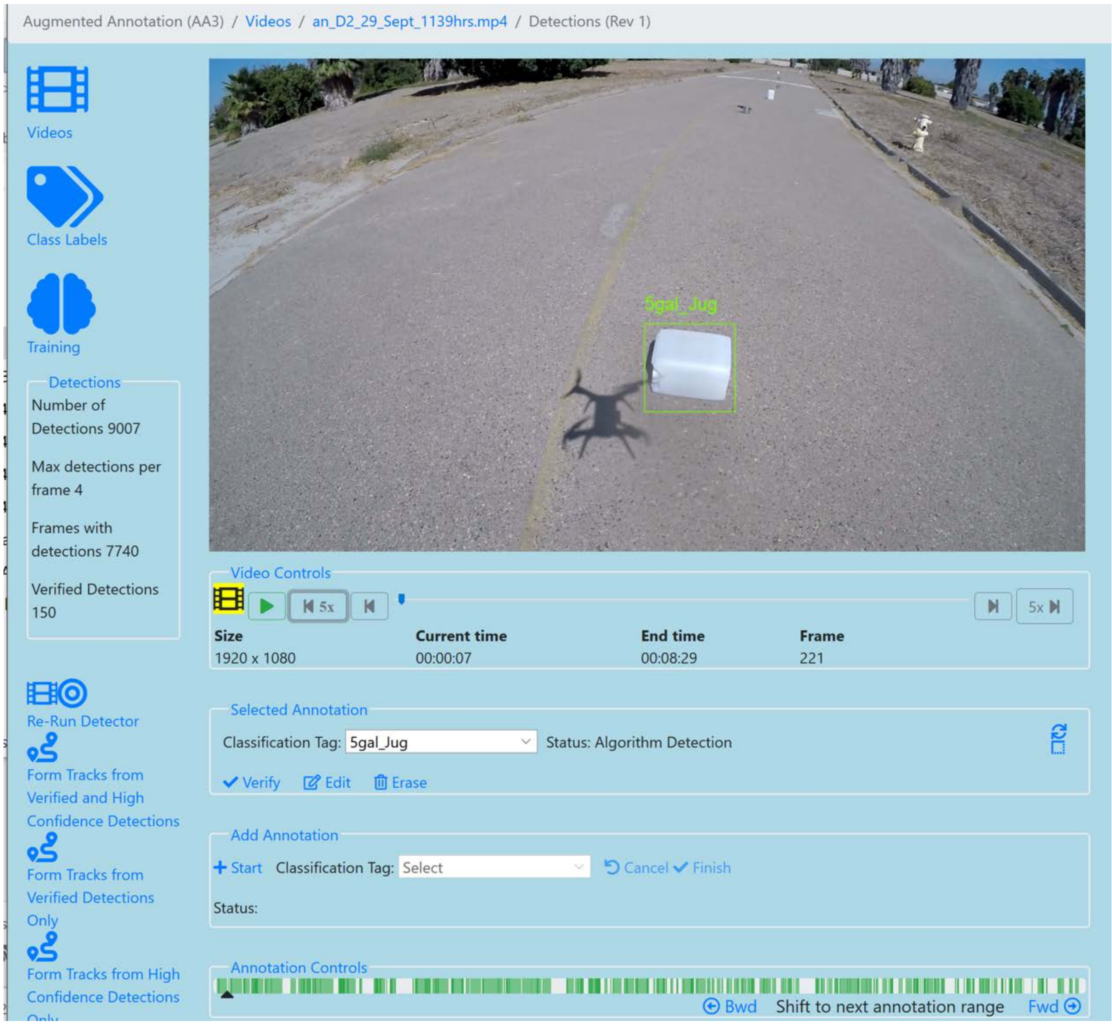
Figure 2. Augmented Annotation components.

Augmented Annotation Phase 3 (AA3) study evaluates the efficiency gain from utilizing the iterative training/annotation cycle. The process from a user's point of view appears as:

1. Apply an object detector that had been trained to new video data.
2. Verify/edit the detections. Only one verified detection is needed for each consecutive appearance of an object instance, though multiple verified detection instances will not negatively impact tracking.
3. Apply tracker to verified detections.
4. Verify tracked annotations and edit if tracks are imperfect.

3.1 USER INTERFACE

The AA3 user interface is implemented in JavaScript. Figure 3 shows a screenshot of the detection verification interface.


Augmented Annotation (AA3) / Videos / an_D2_29_Sept_1139hrs.mp4 / Detections (Rev 1)

Detections

Number of Detections 9007

Max detections per frame 4

Frames with detections 7740

Verified Detections 150

Video Controls

Size	Current time	End time	Frame
1920 x 1080	00:00:07	00:08:29	221

Selected Annotation

Classification Tag: 5gal_Jug Status: Algorithm Detection

Add Annotation

Classification Tag: Select

Status:

Annotation Controls

Re-Run Detector
Form Tracks from Verified and High Confidence Detections
Form Tracks from Verified Detections Only
Form Tracks from High Confidence Detections Only
5x
5x
✓ Verify
Edit
Erase
+ Start
Cancel
✓ Finish
Bwd
Shift to next annotation range
Fwd

Figure 3. AA3 detection verification interface. Users have the ability to edit an automated pre-annotation for class label and bounding box, and verify once satisfied. Users are also presented with detection occurrence information through the time bar on the bottom of the display.

Once a new video is loaded, the user is able to choose an object detector, apply it to the video to obtain the automated pre-annotations, and verify the annotations as appropriate, initiate tracking, then verify the tracks. During detection verification, a user is able to edit object class, object bounding box, and approve a detection as verified. The user is also presented with information on time frames where detections were found. This gives the user awareness for where objects are detected. The user is able to initiate visual tracking to propagate annotations by starting with only user verified annotations, only high confidence detentions, or both types of initializations. Using high confidence (based on detection score) detections requires the user to have a high level of confidence in the performance of the detector. A low performing object detector will likely produce a large number of tracks that require editing during the track verification phase.

During the track verification phase, a user is able to watch a display of tracked annotations over videos to determine the quality of annotations, delete tracks, remove the ends of tracks, and approve a track of annotations when satisfied. The track verification user interface is similar in appearance to the detection verification phase. It also contains a timeline indicating the starting point of all annotation tracks. The user is able to export verified annotations for training the next iteration of object detector. Except for the step to train the object detector, which is managed through an independent process, all other interactions between user interface, computational backend, and database are managed under Django web framework.

3.2 COMPUTATIONAL BACKEND

The Augmented Annotation System improves annotation efficiency by eliminating most of the time needed for labeling objects through visual tracking of a verified annotation, and by reducing the “seek” time to find targets of interest through pre-annotation using an object detector.

3.2.1 Visual Tracking

In Augmented Annotation Phase 1, MDNet, a 2015 Visual Object Tracking challenge winner was chosen as the visual object detector. The detector was computationally very intensive and slow. We chose SiameseFC tracker, a 2017 Visual Object Tracking challenge winner, as the tracker for AA3 implementation because of its computational efficiency and availability of python implementation. One of the critical elements of visual tracking is to automatically terminate tracking when the tracker drifts off target. We found it necessary to apply a feature match algorithm to tracking results to determine tracker failure. ORB feature matching was selected for this application because of its computational efficiency and freely available and reliable implementation. The combination of SiameseFC tracker and ORB matching enabled reliable automatic track terminations in a majority of tracks. The remaining track errors are removed in the track verification step. Visual tracking may be initiated multiple times on one single appearance instance for an object, which could lead to multiple tracks associated with the instance. We apply a merge step that relies on class label agreement and spatial/temporal overlap to ensure a unique track for each object instance. We use GPU-enabled computation in visual track when available.

3.2.2 Object Detection

An effective object detector in the context of an iterative training/annotation cycle needs to be computationally efficient and fast to train and to inference, in addition to having high accuracy at performing the detection task. To that end, we utilized tiny-YOLOv3 as the default detection architecture. In its default configuration, tiny-YOLOv3 subsamples images significantly, which degrades object detection performance on small pixel-count targets. Instead of the standard image size resolutions widely used, we modified tiny-YOLOv3 architecture to use the full image resolution of input data. As a result, detection performance from tiny-YOLOv3 on full resolution images matched the performance using full

YOLOv3 architecture with downsampled image resolution. In addition, tiny-YOLOv3 is much shallower in network depth and hence is faster to train. We use GPU-enabled computations in both training and the inferencing process.

3.3 DATA STORAGE BACKEND

In our early Phase 1 prototype to demonstrate the Augmented Annotation concept, we used a table implemented in Matlab. In the Phase 2 prototype we used SQLite to store database tables on tracks and video sequences and annotations. In AA3 we tested SQLite with tiny-YOLOv3 running in realtime and found SQLite failing to keep up with detection database write frequency. As a result, PostgreSQL was chosen to achieve performance improvement. The collection of tables includes 1, input videos; 2, detectors and versions; 3, tracks with associated video and class labels and verification status; and 4, annotations with class labels, method of acquisition, and verification status. Annotations are exported from the database to xml format along with the associated images for use in training a new iteration of object detector.

4. QUANTIFYING EFFICIENCY

The goal of the Augmented Annotation Phase 3 study is to evaluate and quantify the efficiency gain from using the iterative training and annotation cycle. A more fundamental question in iterative training of the object detector is the selection of datasets that will optimize object detection performance. To that end, a dataset supplied by DoD sponsor was used to perform the evaluations.

The DoD Hackathon data set is mostly videos collected using a small aerial drone ora ground based RC car of outdoor scenes with targets of interest embedded in either an asphalt road surface or on the dirt road shoulders. There are thirteen classes of targets included in the dataset, though only eleven of the classes were used in training the baseline object detector. The videos can be categorized into the following types:

1. 360 degree collection. A single target of interest was placed on an asphalt surface and almost always in view of the camera. Videos were collected from a drone camera flying around the target using various look angles. This is a dataset which we annotated in the Augmented Annotation phase 1 study.

2. Flyby on asphalt. Targets of interest were positioned on an asphalt road surface. Videos were collected from a drone flying above and following the road. Speed may vary between different video samples and camera look angles also change significantly between video sequences.

3. Driveby on asphalt. Targets of interest were positioned on an asphalt road. A camera was mounted to a remote control ground-based vehicle, driven on the road on which targets were positioned. These videos have a significant amount of jitter and can induce nausea in the viewer.

4. Flyby on dirt road. Targets were positioned on the dirt shoulder portion of a road. This introduced a significant change in the imaging conditions on which a detector is expected to function.

5. Indoor data. Targets were positioned in indoor facilities, and videos were collected from an aerial drone. There is insufficient coverage in the number of target classes in this video collection. As a result, we excluded indoor data from consideration.

During the AA1 study, we annotated the 360-degree dataset by presenting the annotator with frames sampled at fixed frequency. This eliminates the need to seek frames for annotation. With the use of visual tracking to propagate initial annotations, we were able to achieve approximately 120 annotations per minute of human effort. In addition to the 360-degree data, we also annotated a number of video sequences where targets do not persist through a video. An annotator needs to watch the video data to seek targets before selecting frames for labeling, thus incurring “seek” time. Under this scenario, the average number of annotations produced by an annotator is 89 annotations per minute of human effort.

The use of an object detector to pre-annotate data has the effect of reducing or eliminating seek time in searching for a target. To investigate the effectiveness of a detector on annotation efficiency gain, we select data that represents each of the video collection categories where there is sufficient coverage of target classes. We selected the following videos for experimentation in AA3:

Table 1. Video Data used in AA3

Video	File Name
Flyby_asphalt1	D3_29_Sept_1332hrs.mp4
Flyby_asphalt2	D2_29_Sept_1139hrs.mp4
Driveby_asphalt	GOPRO151.mp4
Flyby_dirt_road	GOPRO166.mp4

The videos were downsampled to 30 frames/sec and divided into two halves, one of two and two of two, for training the object detector and for validating the detector. We trained four object detectors using different combinations of training data:

Table 2. Detector Versions and their Training and Validation Data

Detector Version	Pre-training	Training Data	Validation Data
V0	MS COCO	360 degree video	Flyby_asphalt1 subset
V1	V0	Flyby_asphalt1 (1/2)	Flyby_asphalt1 (2/2)
V2	V0	Flyby_dirt_road (1/2)	Flyby_dirt_road (2/2)
V3	V0	4K images from each of 360 degree view, Flyby_asphalt 1 of 2, Flyby_dirt_road 1 of 2, Driveby_asphalt	4K images from each of 360 degree view, Flyby_asphalt 2 of 2, Flyby_dirt_road 2 of 2, Driveby_asphalt (excluding training)

The purpose of training detectors with different combinations of data is to demonstrate the significance of training data selection.

To measure the annotation efficiency gain, we use the number of annotations per minute of human effort as one metric. The baseline performance to demonstrate the utility of using the object detector for the purpose of pre-annotation, is the “no-detector-assist” annotation speed of 89 annotations per minute of human effort (AA1 measurement).

Table 3. Increasing Annotation Efficiency with Augmented Annotation Iterative Cycle

Video	Detector Version	Detection Verification Duration (min)	Track Verification Duration (min)	Number of Annotations	Number of Annotations per min	Sec. per Annot.
Flyby comb. from AA1	No detector				89	0.674
Flyby_asphalt1	V0	75	135	39799	189	0.317
Driveby_asphalt	V0	33	26	10881	184	0.325
Flyby_asphalt2	V1	45	58	27257	267	0.225

Detector V0 was trained on 360-degree data which were of targets on an asphalt road. When applied to Flyby_asphalt1, detector V0 performed sufficiently well and most of the targets were detected at some point during each occurrence of the targets. The only challenging targets were extremely small targets such as a key fob or a GoPro camera. Detector V1 was initialized with V0 and additionally trained with annotations from Flyby_asphalt1, which were produced with detector V0 pre-annotating the video. Detector V1 performed even better at pre-annotating Flyby_asphalt2 than V0 did on Flyby_asphalt 1. Even the small targets were detected often. As a result, there was a significant improvement in annotation efficiency. Measured in annotations per minute of human effort, detector V0 produced a 2X improvement in annotation efficiency gain compared to no detector, and detector V1 produced a 3X improvement in efficiency. A significant amount of time saving is a result of reduced “seek” time. Specifically, in applying detector V1 to Flyby_asphalt2 video, a total of 142 verified annotations were used to initialize tracking, of which 22 (15%) were manually generated and 122 were automatically detected and manually verified; whereas in applying detector V0 to Flyby_asphalt1 video, a total of 215 verified annotations were used to initialize tracking, of which 53 (24%) were manually generated and 162 were automatically detected and manually verified. Applying detector V1, which is initialized with detector V0 and trained with additional data with asphalt background, to a new video of targets in asphalt background reduced the manually initialized tracks by 37% and as a result, reduced the average amount of time to produced an annotation by (0.317-0.225)/0.317 = 29%.

When we applied detector V1 on Flyby_dirt_road video, the detection performance was very poor, at less than 1%. This effectively reverts the Augmented Annotation process to the bootstrapping stage—with no detector, and thus no reduction in “seek” time. The effectiveness of the iterative cycle is dependent on the performance of the detector used in each iteration. We decided to investigate the effect of balancing training data during the annotation process. We trained four versions of detectors as listed in Table 2 and tested them on the four videos (Table 1) either singly or in combination to measure the performance of each detector, which would give us an indication of the annotation efficiency gain from each detector. Table 4 shows the detector versions, the test data, and detector performance. Detector V3, which was trained on a combination of data, had the most robust performance.

Table 4. Detector Performance on Single Video and on Combinations of Videos Detector V3, which was trained on a combination of data, had the most robust performance.

Detector Version	Validation/Test Data	mAP
V0	Flyby_asphalt1	66.55%
V1	Flyby_asphalt2	78.27%
V0	Flyby_dirt_road	6%
V1	Flyby_dirt_road	<1%
V2	Flyby_dirt_road	43%
V0	Combo of 4 videos	35.28%
V1	Combo of 4 videos	43.04%
V2	Combo of 4 videos	16.93%
V3	Combo of 4 videos	51%

We ran out of time to perform experiments on further iterations of the annotation/training cycle. However, our exploration into annotation data selection made a more important contribution by determining that blind application of iterative annotation and detector training can negatively impact annotation efficiency. Expertise and effort is needed in selecting data that are representative of application environment in the annotation process, and training needs to be managed to use the training data with distribution of characteristics similar to the application data.

5. LESSONS LEARNED

We have shown through our experiments that the iterative cycle of Augmented Annotation can be highly effective in increasing annotation efficiency. In the best case scenario where a detector is trained with data similar to new data that is to be annotated, by the second iteration annotation efficiency is at three times the Augmented Annotation bootstrapping annotation speed, or 66 times the manual annotation speed. However, if a detector is iteratively trained with a highly biased dataset, its performance could drop significantly. This has the effect of reverting the iterative cycle performance to the bootstrapping performance. One way to counter this negative effect is to deploy a subject matter expert to select wide varieties of data and group them for use in each iterative cycle to ensure that training data does not become overly biased. Another potential method is to use an active learning algorithm which pre-processes all incoming data to determine a suitable dataset to request annotation effort from human annotators.

We learned that human capacity in visual verification of either detections or tracks is limited by either human attention span or nature of sensor noise such as excessive camera motion. Ultimately, if only human verified data is allowed to continue training the next detector, the ability of a human annotator to approve detected or tracked annotation causes a road block when the detector performance becomes sufficiently high. One way to overcome human limitations is to reduce the role of a human annotator in the verification process. Detector training could utilize both user verified annotations and automated annotations, but with different confidence levels.

In addition to lessons learned about the Augmented Annotation process itself, we also noticed a few implementation issues. Synchronization of video streaming play and annotation bounding boxes can be challenging. Our current implementation uses video streaming and relies on the web browser to request video data. The precise frame shown on screen is not precisely know at all times hence the challenge in displaying the correct annotation. The current detection verification process ties detection visualization to video timeframe, which forces the user to wait for the next occurrence of a target. Decoupling detection verification from time will reduce user wait time.

This page intentionally left blank.

6. AUGMENTED ANNOTATION SYSTEM ARCHITECTURE

The Augmented Annotation process showed significant reduction in human effort in generating annotation datasets for training detectors. In the video domain, the process reduces human effort to through two channels: 1, by reducing labeling time through use of visual tracking to propagate annotation labels, and 2, by reducing seek time through applying the object detector to localize target objects. Our experiments in Augmented Annotation Phase 3 demonstrated upwards of 3X improvement in annotation speed with the use of iterative detector training and annotation compared to using visual tracking alone (66X if compared to the manual annotation process). The application of iterative detector training will not by default improve annotation speed, as the selection of data for annotation and training greatly impacts the detector performance. A good detector helps to reduce the seek time in finding video times when an object appears. A low performing detector, resulting from training using biased data, could eliminate the advantage of applying detection to pre-annotated video data by failing to detect a significant portion of the object occurrences. While an expert user can select data to produce an annotation detector that would be representative of application environments, we recommend that an active learning agent pre-processes new data and selects a subset for annotation. For example, an active learning could sample the pool of video collections and apply an existing detector to selected data to start pre-annotating data. The active learning can be applied again to the result of object detection to select data for which a human assigned/verified label would produce maximal effect on the detector. Figure 4 below presents a process and data flow for our recommended Augmented Annotation implementation, including the active learning component.

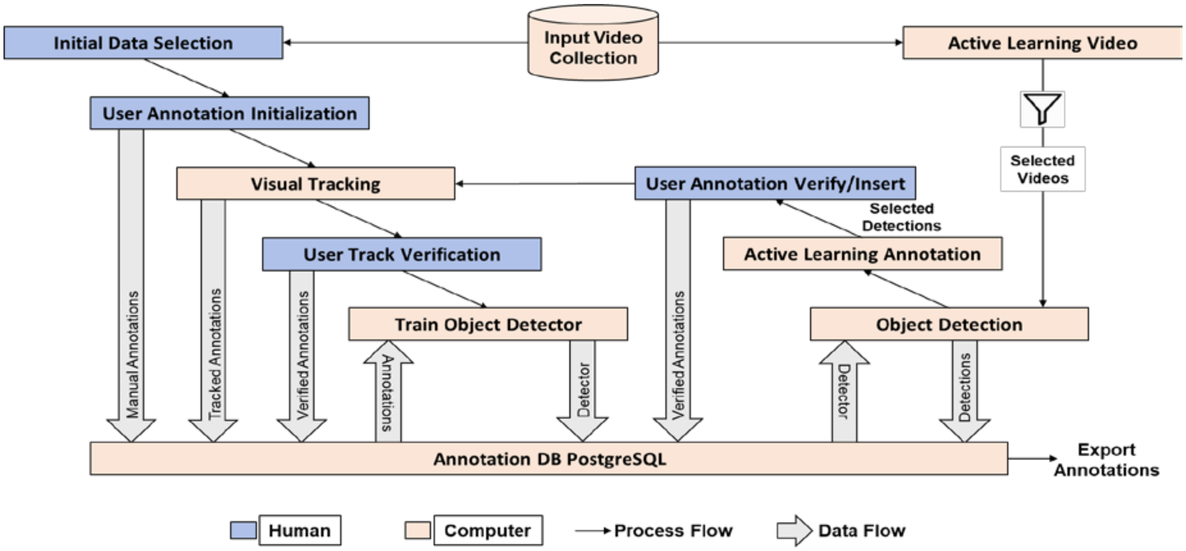


Figure 4. Augmented Annotation process and data flow.

One of the bottlenecks in the Augmented Annotation process is the ability of human users to verify detections/tracks. Current AA3 framework for the training detector utilizes only verified detections, because we had no way of indicating a lower confidence level for non-verified detections. To remove the bottleneck of human verification time, we recommend weakly supervised training, such as the Snorkel framework, which could be used on a combination of user annotated/verified data and machine learning detected annotations.

Some of the implementation details that we would recommend for a deployable Augmented Annotation are to decouple the detector, the visual tracker, and the active learning components so that they may be updated with the latest version of each of these machine learning and computer vision components. This practice would allow for future upgrades to any of these algorithm components, particularly since fields such as machine learning make huge advances on a regular basis. For example, in AA1 we chose MDNet as the default visual tracker, which was replaced with SiameseFC tracker (winner of the 2017 Visual Object Tracker challenge) in AA3. The prototype AA3 implementation integrated the visual tracker tightly, which would make upgrading to the next better performing visual tracker more difficult. Future implementation of Augmented Annotation needs to emphasize modular programming to allow for replacement of any ML/computer vision algorithms.

AA3 user interfaces is web-based and written in JavaScript. We recommend that the deployable system use a similar framework. One of the occasional issues with the AA3 prototype implementation is that synchronization between video and annotations appears to fail occasionally due to the two data sources utilizing different visualization interfaces: the video is streamed using the web browser default video player, the annotation bounding boxes were managed by JavaScript directly. One way to solve the synchronization issues definitively is to convert all videos to image frames and implement a video play that operates on image frames. Depending on level of tolerance on annotation quality, the lag in synchronization issue may or may not be acceptable. Another recommended user interface improvement is to present detection verification in the form of static icon images in addition to the time domain presentation.

REFERENCES

1. Redmon, J., A. Farhadi. “YOLOv3: An Incremental Improvements” https://arxiv.org/abs/1804.02767.

2. Nam, H., H. Bohyung. “Learning Multi-Domain Convolutional Neural Networks for Visual Tracking”. CVPR 2016

3. Bertinetto, L., J. Valmadre, J. Henriques, A. Vedaldi, P. Torr. “Fully-Convolutional Siamese Networks for Object Tracking”. CVPR 2017

4. Ratner, A., S. Bach, H. Ehrenberg, C. Re. “Snorkel: Fast Training Set Generation for Information Extraction”. SIGMOD 2017

This page intentionally left blank.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. **PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**

1. REPORT DATE *(DD-MM-YYYY)* 09-03-2020	**2. REPORT TYPE** Technical Report	**3. DATES COVERED** *(From - To)*

4. TITLE AND SUBTITLE

Augmented Annotation Phase 3

5a. CONTRACT NUMBER
FA8702-15-D-0001

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

L. Lee

5d. PROJECT NUMBER
3094

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

MIT Lincoln Laboratory
244 Wood Street
Lexington, MA 02421-6426

8. PERFORMING ORGANIZATION REPORT NUMBER

TR-1248

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Defense Threat Reduction Agency
8725 John J Kingman Rd
Fort Belvoir, VA 22060

10. SPONSOR/MONITOR'S ACRONYM(S)
DTRA

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release: distribution unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Automated visual object detection is an important capability in reducing the burden on human operators in many DoD applications. To train modern deep learning algorithms to recognize desired objects, the algorithms must be "fed" more than 1000 labeled images (for 55%–85% accuracy according to project Maven - Oct 2017 O6, Working Group slide 27) of each particular object. The task of labeling training data for use in machine learning algorithms is human intensive, requires special software, and takes a great deal of time. Estimates from ImageNet, a widely used and publicly available visual object detection dataset, indicate that humans generated four annotations per minute in the overall production of ImageNet annotations. DoD's need is to reduce direct object-by-object human labeling particularly in the video domain where data quantity can be significant. The Augmented Annotations System addresses this need by leveraging a small amount of human annotation effort to propagate human initiated annotations through video to build an initial labeled dataset for training an object detector, and utilizing an automated object detector in an iterative loop to assist humans in pre-annotating new datasets.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT Unclassified	**b. ABSTRACT** Unclassified	**c. THIS PAGE** Unclassified	Same as report	34	**19b. TELEPHONE NUMBER** *(include area code)*

This page intentionally left blank.