Summarizing and Searching Video

The volume of aerial surveillance video is outpacing the current, manual monitoring capabilities. DoD analysts need capabilities that reduce workload by identifying, summarizing, and creating alerts about critical events and patterns.

Initial Approach

Our initial approach (Figure 1) applied object detection, tracking, and stacked sparse LTSM auto encoders to identify unique segments of video for a summary.

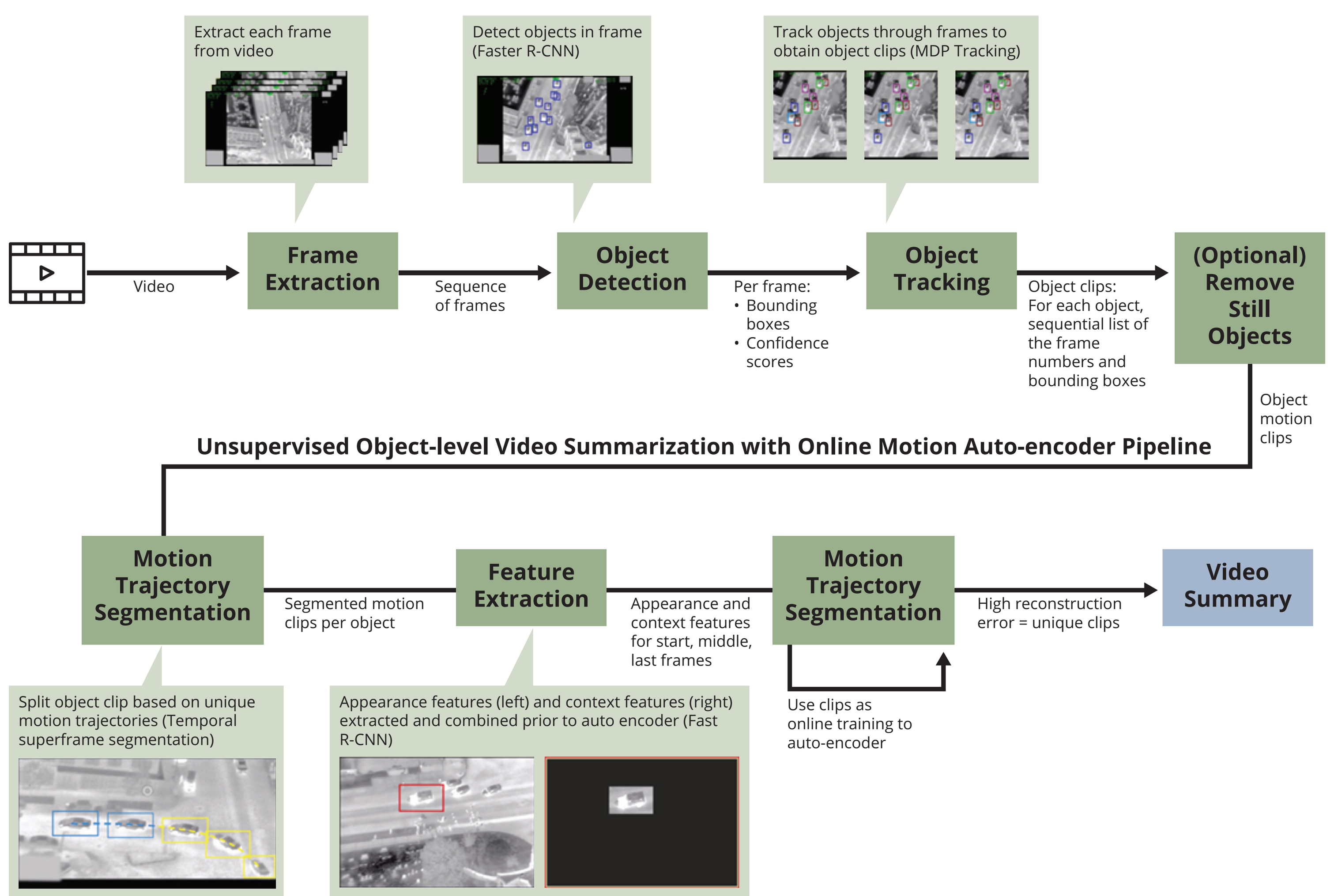

Figure 1: Unsupervised Summarization

This approach proved useful for triaging large volumes of video data to select a subset for further scrutiny, but did not provide useful summaries of the most important events in a video. The result was similar to a movie trailer that provides information useful for deciding whether or not to see the movie, but not for determining the movie's plot.

Based on these results, we revised our approach to better match the needs of an analyst.

Revised Strategy: Pattern of Life Analysis

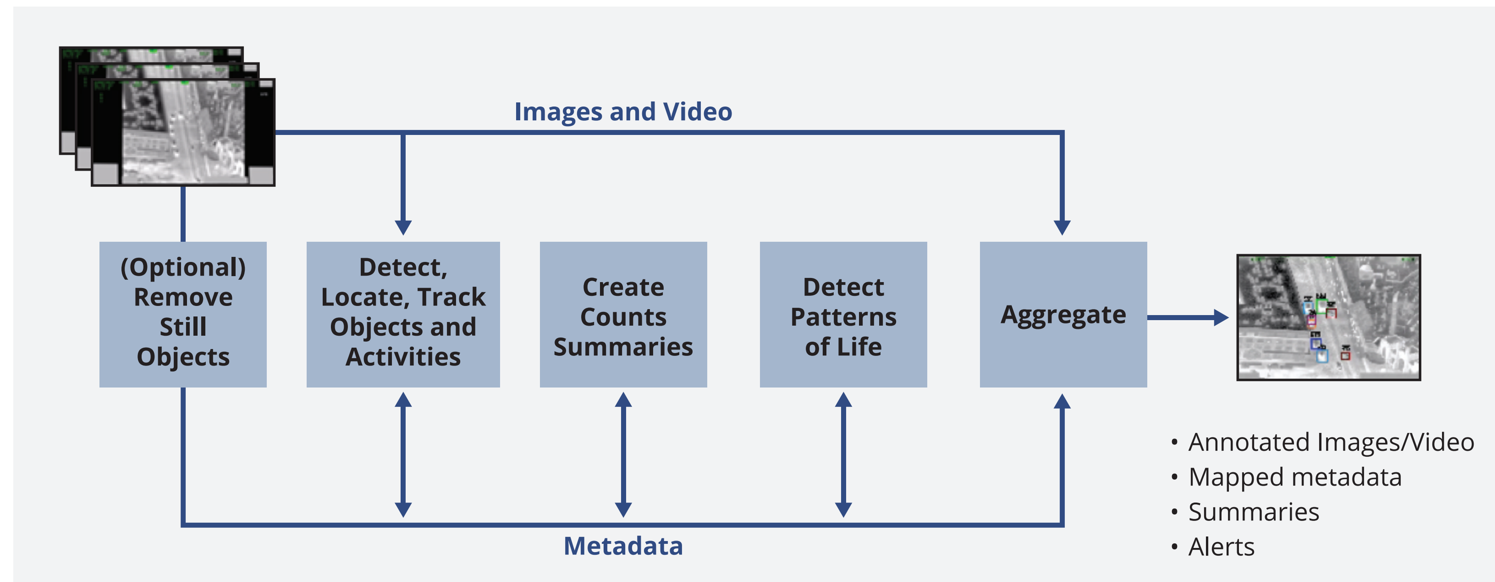

Figure 2: Analysis Pipeline

Our revised strategy focuses on a common surveillance problem of identifying situations of interest relative to a specific area being observed (e.g., a compound). We are:

- selecting and training state-of-the-art classifiers and trackers on images and video representative of DoD aerial surveillance scenarios gathered during these scenarios
- analyzing extracted objects and tracks using statistical and machine learning techniques to summarize data, recognize interactions, and determine patterns of life at the compound

Analysts will be able to:

- set alerts for specific objects, activities, or patterns
- display summaries of detections over time and space
- summaries and alerts will be provided for increasingly complex forms of analysis, from recognition of the signatures of specific objects to prescriptions for suggested courses of action

Example: Anomalous Track Detection

As an initial step toward understanding tracks, we developed a LSTM-Autoencoder algorithm to detect unusual tracks. The algorithm takes as input sequences of spatiotemporal points (tracks) and calculates an anomaly score for each track.

The LSTM-Autoencoder works by learning a compact representation for each track, then attempts to reconstruct that track from the encoding. Behaviors which are under-represented in the training set will be difficult to reproduce and have a high reconstruction error.

Figure 3: Anomalous (Zigzag) Track

To test our algorithm, we created synthetic track data using the SUMO traffic simulator on a set of streets surrounding the SEI (see Figure 3). Each generated track was a best path between a randomly chosen start and stop location. We inserted an additional handed edited track "ZigZag" that was not a best path to see if our algorithm could detect it.

Anomaly Score Distribution

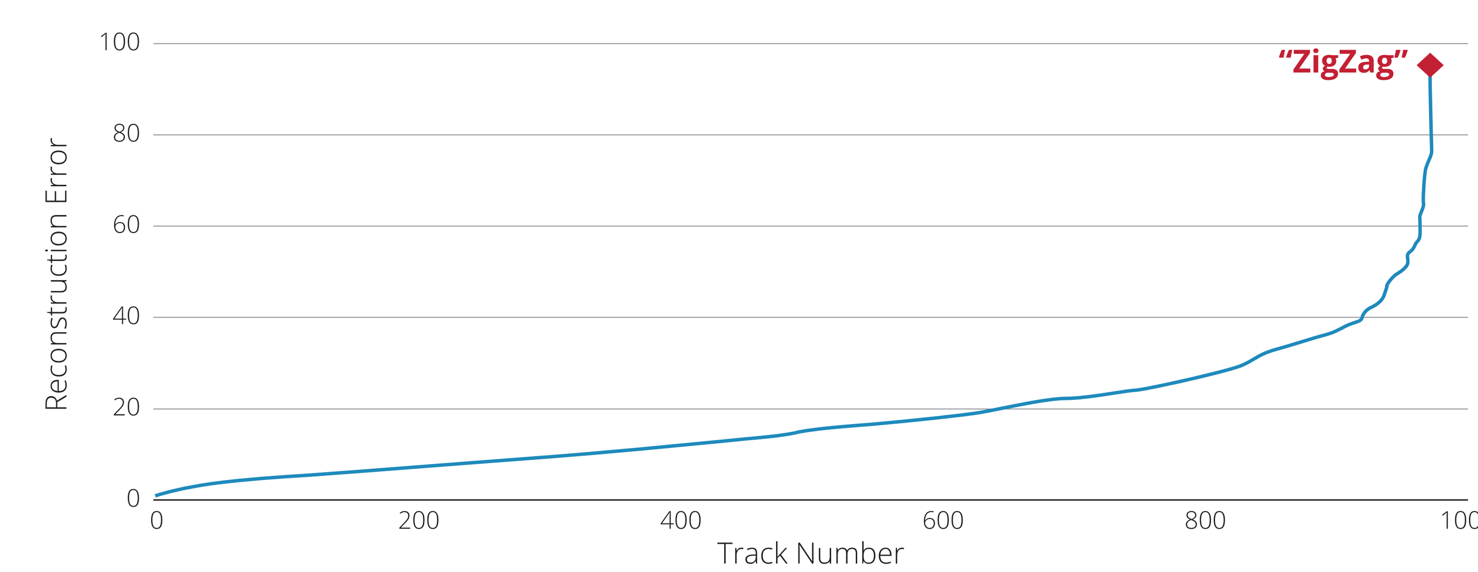

Figure 4: Score Distribution

Figure 4 shows the distribution of score values for all 974 tracks. "ZigZag" track is marked on the graph and had the 2nd highest observed anomaly score. The highest scoring track experienced multiple traffic jams at multiple intersections along its route. Another high scoring (anomalous) track was a vehicle making a u-turn.

Next steps include tuning autoencoder performance and testing against DoD data.

Ed Morris| ejm@sei.cmu.edu

Distribution Statement A: Approved for Public Release; Distribution is Unlimited

Copyright 2018 Carnegie Mellon University. All Rights Reserved.
This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.
The view, opinions, and/or findings contained in this material are those of the author(s) and should not be construed as an official Government position, policy, or decision, unless designated by other documentation.
NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.
[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.
Internal use:* Permission to reproduce this material and to prepare derivative works from this material for internal use is granted, provided the copyright and "No Warranty" statements are included with all reproductions and derivative works.
External use:* This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other external and/or commercial use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.
* These restrictions do not apply to U.S. government entities.
Carnegie Mellon® is registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.
DM18-1128