Actor-Centric Tubelets for Real-Time Activity Detection in Extended Videos

Effrosyni Mavroudi, Prashast Bindal, and René Vidal
Mathematical Institute for Data Science, Johns Hopkins University, Baltimore, MD, USA
{emavrou1@, rvidal@}jhu.edu, bindalprashast@gmail.com

Abstract

We address the problem of detecting human and vehicle activities in long, untrimmed surveillance videos that capture a large field of view. Most existing activity detection approaches are designed for recognizing atomic human actions performed in the foreground. Therefore, they are not suitable for detecting activities in extended videos, which contain multiple actors performing co-occurring, complex activities with extreme spatio-temporal scale variations. In this paper, we propose a modular, actor-centric framework for real-time activity detection in extended videos. In particular, we decompose an extended video into a collection of smaller actor-centric tubelets of interest. Each tubelet is a video sub-volume associated with an actor and includes adaptive visual context for recognizing the actor's activities. Once these tubelets are extracted via an object-detection-based approach, we are able to detect activities in each tubelet by focusing on the actor situated in its foreground. To accurately detect the activities of a tubelet's actor we take into account the interactions with other detected actors and objects within the tubelet. We encode such interactions with a dynamic visual spatio-temporal graph and process it with a Graph Neural Network that yields context-aware actor representations. We validate our activity detection framework on the MEVA (Multiview Extended Video with Activities) dataset and the ActEV 2021 Sequestered Data Leaderboard and demonstrate its effectiveness in terms of speed and performance.

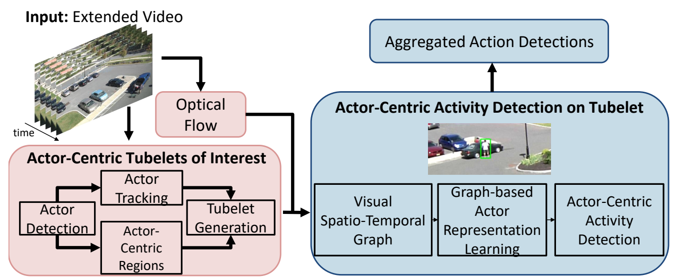

Figure 1: Overview of our proposed actor-centric framework for complex activity detection in extended videos. It consists of two main components: (a) actor-centric tubelet generation and (b) activity detection per tubelet. The first component generates spatio-temporal tubelets of interest, which are associated with a single primary actor (person or vehicle) and capture all the relevant spatio-temporal visual context (scene cues, interacting objects, etc.). The second component predicts the activities performed by an actor over time based on local motion cues (optical flow) and spatio-temporal actor-object interactions. Details for their implementation are provided in Sec. 3.

1. Introduction

As the amount of unconstrained video data gathered daily by surveillance cameras increases, the need for automatic systems that can detect events of interest in security videos is also growing. The majority of such security videos are *extended in time and space* [31, 8], i.e., they are long untrimmed videos that capture multiple actors of various types (people, vehicles) performing multiple activities in various regions of indoor or outdoor scenes.

Powered by deep convolutional networks that process whole video frames and large datasets with rich human annotations, modern video understanding systems are capable of accurately detecting hundreds of human action classes in benchmark datasets [6, 38]. However, many of these popular datasets hide the inherent complexity of action recognition, by either focusing on trimmed videos with a single actor performing a single activity [38] or videos capturing activities performed by a few actors [20, 17, 15], occupying mostly foreground pixels. They also contain only activities performed by humans. The performance of state-of-the-art frameworks is indeed shown to degrade as (a) the number of actors in a scene increases [45], (b) their scale decreases [45], and (c) the complexity of activities increases [15]. Moreover, most activity recognition methods are not suitable for processing extended videos in real time. These limitations affect the ability to deploy these systems for real-time activity detection in extended videos containing a large number of actors (e.g., an average of around 30 actors) of varying types and scales, including tiny actors, performing multiple activities of varying durations [8].

Existing approaches for activity detection in extended videos narrow down the visual search space by identifying video sub-volumes, such as cuboids [14], action tubes [33], or actor tracks [36], that might contain activities. A cuboid is a sequence of bounding boxes with the same spatial coordinates, thus it can be used to crop a valid sub-video and can be fed as input to modern action recognition models. However, the rigid cuboid shape does not necessarily capture the versatile nature of actions. In contrast, action tubes are flexible spatio-temporal sub-volumes capturing relevant spatial contextual cues, but they are typically very short and fail to capture long-term temporal context. Actor tracks are ideal for capturing such temporal context, but might be impractical for real-time activity detection in extended videos for two reasons. First, in typical surveillance videos, such as videos of crowded parking lots, there is a large number of person and vehicle tracks. It is infeasible to process all these tracks under the real-time action recognition constraint. Second, it is not trivial to combine tracks in order to obtain the relevant visual context for detecting various types of activities, such as activities involving a single actor, interaction between actors or actor-object interactions.

In this work, we propose an actor-centric framework for real-time action detection of complex human and vehicle activities of varying spatio-temporal scales in extended surveillance videos. Our framework is composed of two main modules: tubelet generation and temporal activity detection per tubelet. First, we propose an object-detection-based tubelet generation module that decomposes an extended video into a collection of action-agnostic *actor-centric tubelets of interest*. Each actor-centric tubelet consists of an *actor tracklet* and a *context tubelet*. The former is a sequence of bounding boxes of variable size that contain the actor (human or vehicle), and the latter is a sequence of bounding boxes of constant size that captures adaptive, long-range spatio-temporal context for recognizing the activities of that actor. Each actor-centric tubelet is then passed to the second module, which detects the activities performed by an actor over time based on local motion cues (optical flow) and spatio-temporal actor-object interactions. A popular approach for actor-centric action detection applies action classifiers on top of local actor features pooled from an intermediate feature map of a 3D CNN model [15, 10]. However, these local actor features do not capture the rich spatio-temporal interactions of the actor with other actors and objects within the tubelet. We model these interactions with a visual spatio-temporal graph, whose nodes correspond to detected actors and objects in the tubelet and whose edges encode different types of potential interactions, and obtain *context-aware actor features* by applying the recently proposed Visual ST-MPNN [28] on this heterogeneous spatio-temporal graph. Our actor-centric activity detection module is trained only with actor-level supervision, without requiring annotations of relevant objects. Finally, activity detections from all tubelets are aggregated to generate the output set of activity detections for the input video.

In summary, the contributions of this work are three-fold. First, we introduce an *actor-centric framework* for real-time activity detection in extended security videos. Second, we propose an object-detection-based approach for generating action-agnostic actor-centric tubelets of interest that capture an adaptive spatio-temporal context for recognizing the activities of the corresponding actor. This module helps us localize activities in space on an actor-level and also reduces the number of regions that need to be processed in order to detect activities, reducing our overall processing time. Third, we encode spatio-temporal actor-object interactions within each optical flow tubelet with a visual spatio-temporal graph and leverage state-of-the-art Graph Neural Networks [28] for obtaining context-aware, discriminative actor representations. We evaluate the proposed approach on the MEVA (Multiview Extended Video with Activities) dataset [8] and the ActEV21 Sequestered Data Leaderboard and obtain competitive activity detection results compared to published methods in terms of both speed and performance.

2. Related Work

Action Detection in Extended Videos. Most prior work on action detection focuses on long, untrimmed videos with activities performed by a few adult actors. Approaches that temporally detect activities by processing whole frames with convolutional networks, such as the RC3D [46], without determining spatio-temporal regions that might contain activities, have been shown not to be able to handle extended videos [8]. Thus, we focus our brief review of related work on approaches that first identify candidate spatial locations of activities. Activities are either localized per frame by leveraging person detections [10, 40, 45], or are localized via spatio-temporal volumes, like short tubes [12, 34, 37, 23, 18] or tracks [7]. However, these approaches become impractical for detecting activities in extended surveillance videos, not only because they are not able to detect vehicle activities, but also because they will typically result in a large number of proposals, hurting runtime performance.

Detecting complex activities in extended, multi-person videos [31] is a more challenging and computationally demanding task, which requires narrowing down the visual search space by identifying regions that might contain activities. Our proposed approach, that leverages actor tracklets to spatially localize activities, is inspired by early work which tracked moving objects [21, 41] obtained by object detectors [36] or background subtraction [39, 50], and represented those tracks with hand-crafted, global representations. However, we lift simplifying assumptions, such as activities being only human-vehicle interactions and a single activity happening in each region at a time [36], or videos being temporally pre-segmented [1]. Furthermore, we combine actor tracklets with tubelets [22], which allows us to capture adaptive, dynamic spatio-temporal context. Our work is also complementary to recent approaches that employ global deep representations of cuboids [14, 13, 26] or short tubes [33], and offers additional benefits, e.g., modeling of spatio-temporal interactions and long-term temporal context, as well as localization of the actors.

Interaction-based Region Representation Learning. Modeling spatio-temporal interactions between actors and objects has a long history in video understanding [19, 29, 4, 32]. However, most of prior work has focused on modeling interactions between regions with undirected graphical models in a discrete label space [49, 43, 30], where the regions were represented with hand-crafted features. Instead, the focus of our work is to leverage such interactions for learning context-aware actor representations (continuous features). Our activity detection model builds upon recently developed deep architectures called Graph Neural Networks (GNNs) [9], which enable representation learning on graph-structured data. Although GNNs have recently been applied to video understanding [42, 40, 11, 48, 2], they have not been explored for activity detection in extended videos. Our work adapts the Visual ST-MPNN [28], a GNN tailored to representation learning on heterogeneous spatio-temporal graphs, to the task of actor-centric activity detection on tubelets and replaces appearance actor/object features with local motion features.

3. Actor-Centric Activity Detection

This section presents our proposed actor-centric framework for human and vehicle activity detection in extended videos. The overview of our framework is illustrated in Figure 1. An extended video is decomposed into basic units, called actor-centric tubelets of interest. Each tubelet is associated with an actor tracklet and ideally captures all the relevant spatio-temporal visual context (scene cues, interacting objects, etc.) for recognizing the activities of the actor. For the purposes of activity detection in extended surveillance videos, we consider humans and vehicles as actors, since the activities of interest include atomic human activities (e.g., *person closes facility door*), group human activities (e.g., *person embraces person*), human-vehicle interactions (e.g., *person closes trunk*) and atomic vehicle activities (e.g., *vehicle turning left*). Our action recognition module encodes the rich spatio-temporal visual context in spatio-temporal actor-object visual graphs and learns context-aware actor representations with the Spatio-Temporal Message Passing Neural Network (ST-MPNN). In the following, we first define the actor-centric tubelet. Then, we describe in details our approach for (a) actor-centric tubelet generation and (b) supervised temporal multi-label action recognition per tubelet. Finally, we discuss how to post-process the time series of action scores per tubelet in order to output final action detections in the input extended video.

3.1. Actor-Centric Tubelets of Interest

An actor-centric tubelet of interest is defined as a tuple of two bounding box sequences of the same length: (a) an actor tracklet, i.e. a sequence of actor bounding boxes linked by an actor tracker, and (b) a context tubelet, i.e. a sequence of bounding boxes of constant height and width that contain the actor in addition to relevant spatial context. Formally, given an extended video with spatio-temporal dimensions (H, W, T), each actor-centric tubelet, denoted τ_i, is described as: $\tau_i = (t_s^i, t_e^i, \mathcal{B}_a^i, \mathcal{B}_c^i)$, where t_s^i is the start frame, t_e^i is the end frame, $\mathcal{B}_a^i$ is the actor tracklet, and $\mathcal{B}_c^i$ is the context tubelet. Both actor tracklet and context tubelet are sequences of bounding boxes of length $L = t_e - t_s + 1 \leq T$ denoted as $\mathcal{B}_a = [(x_0^a, y_0^a, w_0^a, h_0^a), \ldots, (x_L^a, y_L^a, w_L^a, h_L^a)]$ and $\mathcal{B}_c = [(x_0^c, y_0^c, w^c, h^c), \ldots, (x_L^c, y_L^c, w^c, h^c)]$, respectively, such that for each frame t the actor bounding box is included in the context bounding box and the context boxes have constant height and width, i.e.:

$$0 \leq x_t^c \leq x_t^a < x_t^a + w_t^a \leq x_t^c + w^c \leq W - 1 \quad (1a)$$

$$0 \leq y_t^c \leq y_t^a < y_t^a + h_t^a \leq y_t^c + h^c \leq H - 1 \quad (1b)$$

The actor-centric tubelet of interest has the following desirable properties: (1) it captures long-term temporal context of the actor's actions, since it is associated with an actor tracklet of arbitrary length, (2) it includes long-range spatial context, which complements the actor's appearance for recognizing the actor's activities (since each tubelet can have a different height and width), (3) it defines a valid sub-video with constant height and width, which can be fed to any modern backbone deep neural network for feature extraction, and (4) it can be annotated with unambiguous ground-truth activities at each timestep (given actor-level annotations). We should emphasize that our tubelet is not an action proposal, since it can be associated with zero or multiple actor activities. Rather, it is a sub-volume of interest that is likely to contain activities and is focused on a single actor, similar to videos in most benchmark datasets.

3.1.1 Object-detection-based tubelet generation

Our actor-centric tubelet generation method filters out tracks that are not likely to contain an activity (such as parked vehicles) or are secondary to other actor tracks (such as vehicles involved in person-vehicle interactions). It also determines an adaptive spatial extent for each actor-centric tubelet based on interactions. It achieves this by relying

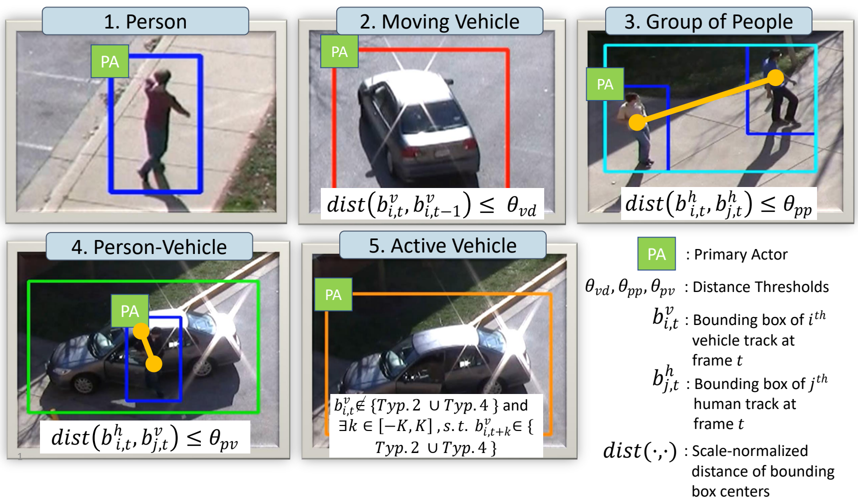

Figure 2: Types of regions of interest with their associated primary actors for sample frames from the VIRAT [31] dataset. For example, regions of Type 5 correspond to recently active vehicles, i.e., vehicles that were (will be) moving or were (will be) associated with an actor detection within a look-back (look-ahead) window of K frames.

only on object detections without requiring training with action spatio-temporal annotations. In particular, it consists of four stages: object detection, actor tracking, actor-centric region of interest extraction, and tubelet generation.

Object Detection. We initialize our tubelet generation pipeline by detecting objects per frame with the Faster R-CNN [16] off-the-shelf object detector, which was trained on the external MSCOCO [25] image dataset.

Actor Tracking. We track detections from each actor class (person or vehicle) using the SORT [3] off-the-shelf tracker, which predicts a trajectory using a Kalman filter and matches tracks to detections using a simple IoU metric. Tracking not only provides the basis for linking regions of interest across time to generate actor-centric tubelets, but also helps fill in missing object detections.

Actor-Centric Region of Interest Extraction. The goal of this step is to (a) find actors at each frame that are likely to be involved in activities and (b) identify other actors they might be interacting with. This information will be used to filter out track segments that are not likely to contain activities, such as static vehicles without any people in their vicinity, thus reducing the number of regions fed to our activity detection module with minimal impact on the recall. It will also aid in determining the adaptive, spatial context that is relevant for recognizing the activities of each actor. We use a rule-based approach to find Regions of Interest (ROIs) per frame, where each region corresponds to one out of 5 potential types of ROIs and is associated with a primary actor detection. Such regions are automatically extracted from actor detections by associating them with hand-crafted rules based on scale-normalized distance thresholds. An intuitive illustration of the five types of actor-centric RoIs and their corresponding primary actors, as well as the rules used for their construction, is shown in Figure 2.Note that an actor detection can be the primary actor of zero, one or multiple actor-centric RoIs. For example, a person can be associated with multiple nearby people and vehicles.

Tubelet Generation. Given the actor-centric RoIs extracted per frame, we are now ready to describe the generation of actor-centric tubelets. First, we construct a context bounding box for each actor detection that is the primary actor of at least one actor-centric RoI. This context box is constructed by computing the union of all RoIs which have this actor as their primary actor. Leveraging the extracted actor tracks, context bounding boxes associated with the same primary actor instance are linked over time to construct an actor-centric tubelet of interest, with the sequence of context boxes generating the *context tubelet* $\mathcal{B}_c$, and the sequence of primary actor bounding boxes generating the *actor tracklet* $\mathcal{B}_a$. We would like to emphasize that in contrast to track-based methods, our actor-centric tubelets do not necessarily include a whole actor track, but only track segments that contain actor detections that are primary actors of RoIs. For example, instead of predicting activities for each timestep of a tracked vehicle, we only predict activities for the temporal segments that this vehicle is either moving or people are about to enter/exit it. Still, all detections of this vehicle can serve as context for other tubelets.

Context Tubelet Post-processing. The generated context tubelets might have an irregular shape with sudden changes in the size of the consecutive bounding boxes, e.g., because the number of interacting actors varies with time or because of errors in the association of actors due to occlusions. To alleviate this issue, we enlarge each context bounding box of the tubelet so that they have the same height and width, with its dimension being determined by the largest bounding box of the tubelet. A final refinement step ensures that the tubelet consists of a smoother sequence of context bounding boxes. In particular, a Savitzky-Golay [35] filter is used to estimate smoothed values of the bounding box centers. Then, the top-left context bounding box coordinates are updated accordingly without modifying the tubelet dimensions.

3.2. Actor-Centric Activity Detection on Tubelet

Once an extended video is decomposed into a set of actor-centric tubelets of interest, our system seeks to temporally detect the activities performed by the actor of each tubelet. Our proposed structured activity detection model builds upon the Visual ST-MPNN [28]. It encodes spatio-temporal interactions between actors and objects in a visual graph and learns graph-structure-aware actor embeddings that can be used to recognize activities.

Visual Spatio-temporal Graph. Let $\tau = (t_s, t_e, \mathcal{B}_a, \mathcal{B}_c)$ be an extracted tubelet with length $L = t_s - t_e + 1$. We represent it with a visual spatio-temporal, attributed graph $G = (\mathcal{V}, \mathcal{E})$, which consists of a set $\mathcal{V}$ of actor nodes and object nodes, and a set of edges $\mathcal{E}$. Actor nodes correspond to the bounding boxes of the primary actor tracklet $\mathcal{B}_a$ of the tubelet, while object nodes correspond to other object detections within the context tubelet $\mathcal{B}_c$, including other visible humans and vehicles. The graph is built by adding directed, typed edges that connect nodes. In particular, an edge connecting node j to node i is associated with an edge type ϵ_{ij}. There are three possible edge types: *object-to-actor spatial* ($\epsilon_{ij} = 0$) and *actor-to-object spatial* ($\epsilon_{ij} = 1$) edges connect actor and object nodes in the same frame, while *actor-to-actor temporal* ($\epsilon_{ij} = 2$) edges connect actors across frames. We constrain temporal edges to connect only nodes of the same type between consecutive frames. All graph node attributes $\mathbf{h}_i^{(0)}$ are initialized with ROI-pooled features from a feature map that is obtained by passing a cropped optical flow tubelet through a flow I3D network [6]. Similarly, edge attributes $\mathbf{h}_{ij}^{(0)}$ are initialized with the relative spatial location of the connected nodes.

Graph-based Actor Representation Learning. Given the input visual st-graph, the ST-MPNN iteratively refines the local node and edge features with spatio-temporal contextual cues. Specifically, at each iteration r, the Visual ST-MPNN: (1) computes scalar visual edge weights using edge-type-specific attention mechanisms; (2) computes a message $m_{ij}^{(r)}$ along each edge (i, j) using the attention-based scalar edge weight, the features of the connected nodes and the edge feature; (3) updates the feature of every node by aggregating messages from incoming edges with an update function U; and (4) updates the feature of every

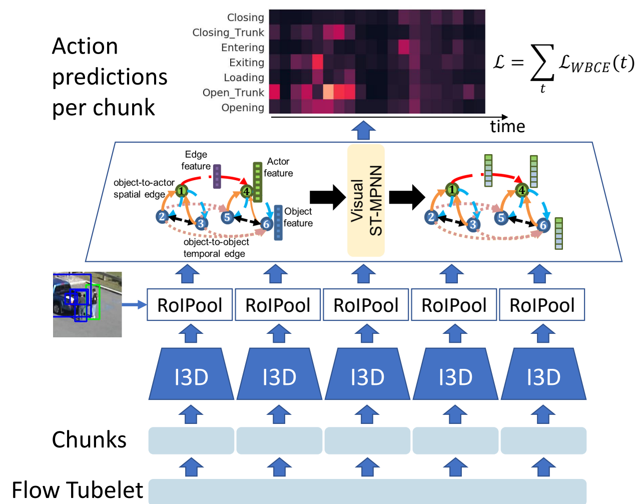

Figure 3: Graph-based activity detection module for actor-centric activity detection in a tubelet.

edge by using the message that was computed alongside it. Importantly, the message passing functions, $M(\cdot)$, are parameterized with learnable weights $W_{\epsilon_{ij}}$ that depend on the edge type ϵ_{ij},

$$\mathbf{m}_{ij}^{(r)} = M(\mathbf{h}_i^{(r-1)}, \mathbf{h}_j^{(r-1)}, \mathbf{h}_{ij}^{(r-1)}; W_{\epsilon_{ij}}) \quad (2)$$

$$\mathbf{h}_i^{(r)} = U(\mathbf{m}_{ij}^{(r)}, \mathbf{h}_i^{(r-1)}). \quad (3)$$

More details about the implementation of the message passing and update functions can be found in the original paper [28]. After R layers of the spatio-temporal MPNN (or equivalently R rounds of node and edge updates), we obtain refined, visual context-aware node and edge features.

Temporal Activity Detection. Let $\mathbf{x}_t$ be the context-aware node feature that corresponds to the tubelet's primary actor bounding box at time t. A linear classifier is applied on $\mathbf{x}_t$ to predict scores for C action classes at time t:

$$\tilde{\mathbf{y}}_t = W_{cls}\mathbf{x}_t + \mathbf{b}_{cls} \in \mathbb{R}^C, \quad t = 1, ..., L, \quad (4)$$

where $W_{cls} \in \mathbb{R}^{C \times d}$ and $\mathbf{b}_{cls} \in \mathbb{R}^C$ are learnable parameters. Since an actor might be performing multiple activities at the same time, we treat the problem as a multi-label per-frame action classification problem, passing scores $\tilde{\mathbf{y}}_t$ through a sigmoid activation function to yield final action probabilities $\hat{\mathbf{y}}_t \in \mathbb{R}^C$.

The output of the previous step is a sequence of probabilities for each activity $a \in \{0, \dots, C-1\}$ for each tubelet timestep t. To obtain final temporal detections for activity a within the tubelet, we need to convert the action scores sequence to a set of temporal segments with start, end times and associated confidence scores. To achieve this, we first apply a median filter to smooth the action prediction probabilities $[\hat{y}_0^a, \dots, \hat{y}_{L-1}^a]$. We then initialize activity detections at the local maxima of the smoothed action score time-series $[s_0^a, \dots, s_{L-1}^a]$. The temporal boundaries of an activity detected at local maximum location t_k, with score $s_{t_k}^a$, are extended by including previous and future timesteps until their action score falls below a relative threshold $\theta \cdot s_{t_k}^a$, where $\theta < 1$ is a hyperparameter. In this way, we can detect activities of arbitrary lengths and can handle several instances of the same activity performed by the tubelet's primary actor, such as consecutive *turning left* activities corresponding to the same vehicle tracklet. We assign the maximum action score of the timesteps included in each action detection as the detection's confidence score. Our system's output consists of action detections that are aggregated from all tubelets.

Training. Our actor-centric activity detection module is trained with actor-level annotations associated with the primary actor of each tubelet. Given the ground-truth activity annotations for the primary actor of a tubelet, the ST-MPNN network is trained jointly with the action classifiers by using a Weighted Binary Cross-Entropy (WBCE) loss per class:

$$\mathcal{L}_{WBCE}(y_t^a, \hat{y}_t^a) = \beta_a y_t^a \log \hat{y}_t^a + (1 - y_t^a) \log(1 - \hat{y}_t^a), \quad (5)$$

where $y_t^a \in \{0, 1\}$ is the ground-truth label for timestep t and action a, and $\hat{y}_t^a \in [0, 1]$ is our model's prediction. To handle the class imbalance, we apply a weighting factor β_a to positive examples of each class a, which is determined based on the inverse class frequency.

4. Experiments

4.1. Datasets

We validate our method on the MEVA dataset and the ActEV 2021 Sequestered Data Leaderboard. The **MEVA dataset** [8] consists of 5-minute videos capturing indoor and outdoor scenes. There is an ongoing effort for annotating MEVA videos with actor-level annotations of 37 activity classes by Kitware and the community. We use Kitware annotations for 784 of these videos for training our activity detection module and 172 for constructing an internal validation set for our ablation studies. The **ActEV 2021 SDL** [1] consists of sequestered surveillance videos, which are not publicly available. Evaluating a method on this dataset requires submitting an activity recognition system that is compatible with the ActEV Command Line Interface (CLI) protocol and temporally detects instances of 37 activites. The submitted system is then executed on test servers provided by NIST and scores are reported on the public leaderboard.

4.2. Metrics

The activity detection performance of our system is evaluated with the official metrics of the ActEV SDL evaluation: (a) the probability of missed detection at fixed time-based false alarm per minute (Pmiss@0.02tfa), and partial area under the Detection Error Tradeoff curve (nAUDC@0.2tfa). These metrics are calculated by finding correspondences between system activity detections and ground-truth activity instances, where a ground-truth activity instance is considered to be missed if it does not overlap with a system detection for at least one second. For achieving a good performance under these metrics, our system needs to accurately detect activities, while at the same time it needs to minimize the Time-based False Alarm (TFA), which is the proportion of time the system detected an activity when there was none. We used the official scorer [2] for evaluating the system on our MEVA validation set. It computes metrics per video and we report their average.

4.3. Implementation Details

Tubelet generation. Our actor detections correspond to Person and Vehicle (bicycle, car, motorcycle, bus, truck) object detections with confidence score above 0.5. The SORT tracker [3] is used to separately track people and vehicles. Tracks are terminated after not being associated with an actor detection for 64 frames. Afterwards, regions of interest are identified in each frame by associating actor detections with hand-crafted rules, which are based on thresholds of scale-normalized distances: $\theta_{pp} = 6000$, $\theta_{pv} = 5000$, $\theta_{vv} = 500$, and an active vehicle look-ahead/look-back window of 256 frames.

Activity Detection Module. For activity detection on each tubelet, we first crop the tubelets from an optical flow representation of the input extended video. Optical flow is extracted from resized and downsampled RGB frames with the TVL1 algorithm following the same setup as in [13]. To build the visual graph, we first apply an optical flow I3D network [6], which was trained for action classification on MEVA cuboids and shared by the authors of [13], on consecutive 2-second non-overlapping chunks of the input flow tubelet. In this way, we obtain a feature map with a temporal stride of 8 frames for each chunk. We then instantiate the graph on top of the primary actor detections and 10 most confident object detections (with score above 0.1) at the corresponding tubelet frames. Note that we store the centre coordinates of all object detections for a frame of the original extended video in a KD-tree data structure, which enables efficient rectangle range queries. We can then efficiently retrieve all object detections whose centre lies within a tubelet bounding box at a given frame. The initial node features for actors/objects are pooled from the Mixed 4f 3D feature map of the flow I3D for each detected region using RoIAlign [16]. These features are refined to include context by performing 3 rounds of node/edge refinement with

[1] https://actev.nist.gov/sdl

[2] https://github.com/usnistgov/ActEV_Scorer

the Visual ST-MPNN [28], resulting in context-aware 512-dimensional embeddings of actor regions that are fed to action classifiers. The action detection threshold θ is set to 0.8 and median window size is 25 frames (3 chunks).

Training. We jointly train the Visual ST-MPNN and action classifiers on 7151 tubelets extracted from MEVA training videos for 150 epochs using a batch size of 10 tubelets (with a maximum length of 30 seconds). Given ground-truth actor-level annotations, we assign a ground-truth activity to the primary actor of a tubelet at a given frame if its detected bounding box overlaps with the corresponding ground-truth actor with $IoU > 0.5$. We use the Adam [24] optimizer, with an initial learning of $1e^{-4}$.

CLI System. The system submitted to the ActEV SDL is customized to run on a hardware consisting of 4 GPUs with 128GB RAM. It is implemented as a pipeline consisting of several stages with each stage producing an output to be used by the later stages. The stages can be enumerated as follows: 1) Optical Flow Extraction 2) Object Detection and Actor Tracking 3)Tubelet Generation 4) I3D Feature Extraction 5) ST-MPNN Processing. Each stage is parallelizable and spawns several subprocesses/workers which work on multiple videos/chunks simultaneously. Among the stages, stage 3 is CPU-intensive and the rest are GPU-intensive. The pipeline processes the entire test set in batches of 96 videos. Each stage maintains a processing queue of 96 videos and any idle workers consume videos from this queue until the entire video batch has been processed. The number of workers for each of the stages are: 48, 24, 96, 8, and 8 respectively. In all stages except stage-3, we are limited by the GPU memory and hence cannot increase the number of workers anymore. The submitted system slightly differs from the system evaluated on our internal validation set: (a) the object detector is applied on the video with a stride of 4 frames for faster processing, while repeating the bounding box detections in between to accommodate for the skipped frames, and (b) we keep at most 200 actor-centric tubelets from each input video, after ranking them based on motionness cues.

4.4. Experimental Results

Comparison with the state of the art. Table 1 compares the activity detection performance of our method with recently published work and other submitted systems on the ActEV 2021 SDL Known Facility Leaderboard [3]. Our actor-centric framework for real-time activity detection achieves activity detection performance that is close to other published methods [33, 13, 47] (rows 1-3). Specifically, it achieves a nAUDC metric of 48% on the challenging sequestered dataset, while performing at 0.97 real-time. Notably, it achieves this metric despite only training the

[3] `https://actev.nist.gov/sdl#tab_leaderboard`

System	nAUDC	pmiss@0.02tfa	Rel. Time
Cuboids [13]	0.476	-	0.725
Gabriella [33]	0.438	-	0.362
Dense Prop. [47]	0.423	-	-
CMU-DIVA	**0.163**	**0.3424**	0.413
UCF	0.232	0.3793	0.751
UMD	0.262	0.4544	0.380
IBM-Purdue	0.281	0.4942	0.631
Visym Labs	0.283	0.4620	0.721
UMD-Columbia	0.305	0.4716	0.516
UMCMU	0.323	0.5297	0.464
Purdue	0.332	0.5853	0.131
BUPT-MCPRL	0.799	0.9281	**0.123**
MINDS_JHU (Ours)	0.483	0.6649	0.967

Table 1: Temporal detection results on the ActEV 2021 Known Facility SDL as of November 1st 2021. We report the Pmiss@0.02tfa and nAUDC@0.2tfa metrics. Lower nAUDC and pmiss values indicate a superior performance since they are related to missing an activity.

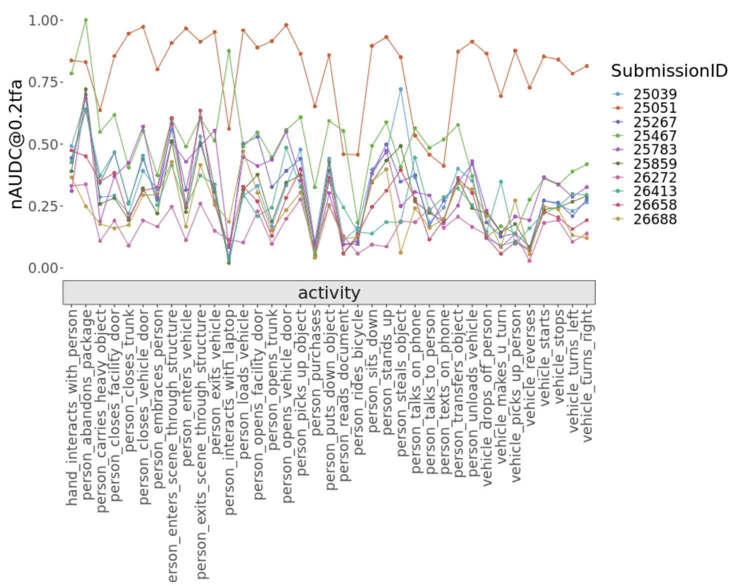

Figure 4: Per-class nAUDC scores for systems on the ActEV 2021 SDL. Our system ID is 25467 (light green).

GNN and action classifiers of our framework using actor-level annotations, in under 3 hours using a single Titan XP GPU (given the extracted visual graph), while relying on off-the-shelf, pretrained networks for object detection and flow feature extraction. When compared to recent system submissions on the ActEV Challenge, which might utilize additional training datasets, end-to-end training, and model ensembles, our system lags behind most of them. However, as we can see in Fig. 4, our system (ID: 25467) performs on par with other methods for a wide range of activities, such as person-vehicle interactions (*vehicle drops-off person*) and vehicle activities (*vehicle u-turn*), while performing significanlty worse on *person abandons package* and *person interacts with laptop*. Our overall performance could be improved by including more samples of these activities in our training set and by fine-tuning our object detector on surveillance data. Furthermore, the I3D could be fine-tuned jointly with the ST-MPNN.

Ablation analysis. We now discuss a variety of ablation studies of different components of our framework. In Table 2, we compare the total number of actor regions that are included in actor tracks with the number of regions that are the primary actors of our actor-centric tubelets. As we can see, our tubelet generation method prunes a large number of tracked actor detections that are unlikely to be performing activities and only feeds 37% of the actor regions to the activity detection module. This helps our model perform real-time activity detection. Despite pruning a large number of actor regions, our generated tubelets retrieve a large number of ground-truth activities (around 80%), as shown in Table 3. The primary cause for missed activity detections are object detection failures of the off-the-shelf, pretrained object detector. In Table 4, we first experiment with two different action classification models to determine the best architecture for our system. In particular, we compare our context-aware feature obtained by applying the Visual ST-MPNN on our visual graph with a baseline feature that is obtained from locally-extracted actor features passed through a trainable two-layer Multi-layer Perceptron of hidden size 1024. Refining the local actor features with the GNN improves performance, verifying our intuition that spatio-temporal actor-object interactions are crucial for detecting activities. Furthermore, we compare generating activity detections of fixed duration (6 seconds) around each local maximum of the score time-series per activity class, instead of adaptively extending the detection to the past and future by a relative score threshold. Surprisingly, fixed duration activity detections lead to a better performance. This can be attributed to the employed detection metrics, which consider a ground-truth activity to be detected as long as 1 second of it overlaps with a system detection.

Tubelet type	Nb. Actor RoIs
Tracks	6783972
ACToIs	**2553404**

Table 2: Impact of tubelet generation method on the number of actor regions that are fed to the activity detection module. *Tracks*: baseline tubelets spanning each actor track of an extended video. *ACToIs*: our proposed Actor-Centric Tubelets of Interest. Results are reported on our internal validation set of Kitware-annotated MEVA videos.

	R@30	R@8	R@1
ACToIs (train)	69.0	81.7	85.0
ACToIs (val)	67.9	81.5	84.0

Table 3: Activity recall of our proposed Actor-Centric Tubelets of Interest on our MEVA training and validation sets. Recall $R@T$ is computed by considering an activity instance as retrieved when at least one tubelet's primary actor overlaps with the ground-truth actor with $\mathrm{IoU} > 0.5$ for at least T consecutive frames.

Feat	Duration	nAUDC	pmiss@0.04	pmiss@0.02
Local	Dynamic	0.558	0.556	0.680
Context-Aware	Dynamic	0.531	0.501	0.663
Context-Aware	Fixed	**0.492**	**0.469**	**0.565**

Table 4: Ablation experiments on our internal MEVA validation set for different design choices of the activity detection module. *Local Actor Feat*: baseline approach that recognizes actor activities based on locally-extracted actor features. *Context-aware Actor Feat*: our proposed approach that learns context-aware actor features with the Visual ST-MPNN. *Dynamic Duration*: generating activity detections of varying durations. *Fixed Duration*: generating activity detections of fixed duration (6 sec) around local maxima.

5. Conclusion

In this paper we introduced an actor-centric framework for detecting complex human and vehicle activities of varying spatio-temporal scales in extended surveillance videos. Our system decomposes an extended video into a collection of actor-centric tubelets of interest, which capture long-range spatial and temporal context for an actor. It then predicts the activities performed by an actor over time based on local motion cues (optical flow) and spatio-temporal actor-object interactions. The modular design of our system makes it amenable to improvements. The current off-the-shelf object detection, tracking and feature extraction backbones can be easily replaced by state-of-the-art networks, such as DETR [5], Joint Detection and Embedding (JDE) multiple-object tracker [44], and TANet [27], respectively. Furthermore, they can be additionally fine-tuned on surveillance data. We leave such improvements to future work.

Acknowledgements. The authors thank Carolina Pacheco Oñate, Ambar Pal and the anonymous reviewers for their valuable comments. We also thank Joshua Gleason and Carlos D. Castillo for providing us with their code and trained I3D model. This research was supported by the IARPA DIVA program via contract number D17PC00345.

References

[1] Mohamed R. Amer and Sinisa Todorovic. Sum-product networks for modeling activities with stochastic structure. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[2] Anurag Arnab, Chen Sun, and Cordelia Schmid. Unified graph structured models for video understanding. In *IEEE International Conference on Computer Vision*, pages 8117–8126, October 2021.

[3] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *IEEE International Conference on Image Processing*, pages 3464–3468, 2016.

[4] William Brendel and Sinisa Todorovic. Learning spatiotemporal graphs of human activities. In *IEEE International Conference on Computer Vision*, 2011.

[5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229, 2020.

[6] J. Carreira and A. Zisserman. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4724–4733, 2017.

[7] Guilhem Chéron, Anton Osokin, Ivan Laptev, and Cordelia Schmid. Modeling spatio-temporal human track structure for action localization. *CoRR*, abs/1806.11008, 2018.

[8] Kellie Corona, Katie Osterdahl, Roderic Collins, and Anthony Hoogs. Meva: A large-scale multiview, multimodal video dataset for activity detection. In *IEEE Winter Conference on Applications of Computer Vision*, 2021.

[9] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural Message Passing for Quantum Chemistry. In *International Conference on Machine learning*, pages 1263–1272, 2017.

[10] Rohit Girdhar, João Carreira, Carl Doersch, and Andrew Zisserman. A better baseline for AVA. *CoRR*, abs/1807.10066, 2018.

[11] Rohit Girdhar, Joao Carreira, Carl Doersch, and Andrew Zisserman. Video action transformer network. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[12] Georgia Gkioxari and Jitendra Malik. Finding action tubes. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 759–768, 2015.

[13] Joshua Gleason, Carlos D. Castillo, and Rama Chellappa. Real-time detection of activities in untrimmed videos. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) Workshops*, March 2020.

[14] Joshua Gleason, Rajeev Ranjan, Steven Schwarcz, Carlos D. Castillo, Jun Cheng Chen, and Rama Chellappa. A proposal-based solution to spatio-temporal action detection in untrimmed videos. In *IEEE Winter Conference on Applications of Computer Vision*, 2019.

[15] Chunhui Gu, Chen Sun, David A. Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, Cordelia Schmid, and Jitendra Malik. AVA: A video dataset of spatio-temporally localized atomic visual actions. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[16] K. He, G. Gkioxari, P. Dollar, and R. Girshick. Mask R-CNN. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2018.

[17] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. ActivityNet: A large-scale video benchmark for human activity understanding. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[18] Rui Hou, Chen Chen, and Mubarak Shah. Tube convolutional neural network (T-CNN) for action detection in videos. In *IEEE International Conference on Computer Vision*, 2017.

[19] T. Huang, D. Koller, J. Malik, G. Ogasawara, B. Rao, S. Russell, and J. Weber. Automatic symbolic traffic scene analysis using belief networks. In *Proceedings of the National Conference on Artificial Intelligence*, volume 2, 1994.

[20] Haroon Idrees, Amir R. Zamir, Yu Gang Jiang, Alex Gorban, Ivan Laptev, Rahul Sukthankar, and Mubarak Shah. The thumos challenge on action recognition for videos "in the wild". *Computer Vision and Image Understanding*, 155, 2017.

[21] Yuri A. Ivanov and Aaron F. Bobick. Recognition of visual activities and interactions by stochastic parsing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22, 2000.

[22] M. Jain, J. C. van Gemert, H. Jegou, P. Bouthemy, and C. G. M. Snoek. Tubelets: Unsupervised action proposals from spatiotemporal super-voxels. *International Journal of Computer Vision*, 2017.

[23] Vicky Kalogeiton, Philippe Weinzaepfel, Vittorio Ferrari, and Cordelia Schmid. Action tubelet detector for spatio-temporal action localization. In *IEEE International Conference on Computer Vision*, Oct 2017.

[24] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, 2014.

[25] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755, Cham, 2014. Springer International Publishing.

[26] Wenhe Liu, Guoliang Kang, Po Yao Huang, Xiaojun Chang, Lijun Yu, Yijun Qian, Junwei Liang, Liangke Gui, Jing Wen, Peng Chen, and Alexander G. Hauptmann. Argus: Efficient activity detection system for extended video analysis. In *IEEE Winter Conference on Applications of Computer Vision Workshops*, 2020.

[27] Zhaoyang Liu, Limin Wang, Wayne Wu, Chen Qian, and Tong Lu. Tam: Temporal adaptive module for video recognition. In *IEEE International Conference on Computer Vision*, pages 13708–13718, October 2021.

[28] E. Mavroudi, B.B. Haro, and R. Vidal. Representation learning on visual-symbolic graphs for video understanding. In *European Conference on Computer Vision*, volume 12374 LNCS, 2020.

[29] Darnell J. Moore, Irfan A. Essa, and Monson H. Hayes. Exploiting human actions and object context for recognition

tasks. In *IEEE International Conference on Computer Vision*, volume 1, 1999.

[30] Nandita M. Nayak, Yingying Zhu, and Amit K. Roy Chowdhury. Hierarchical graphical models for simultaneous tracking and recognition in wide-area scenes. *IEEE Transactions on Image Processing*, 24, 2015.

[31] Sangmin Oh, Anthony Hoogs, Amitha Perera, Naresh Cuntoor, Chia Chih Chen, Jong Taek Lee, Saurajit Mukherjee, J. K. Aggarwal, Hyungtae Lee, Larry Davis, Eran Swears, Xioyang Wang, Qiang Ji, Kishore Reddy, Mubarak Shah, Carl Vondrick, Hamed Pirsiavash, Deva Ramanan, Jenny Yuen, Antonio Torralba, Bi Song, Anesco Fong, Amit Roy-Chowdhury, and Mita Desai. A large-scale benchmark dataset for event recognition in surveillance video. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[32] A. Prest, V. Ferrari, and C. Schmid. Explicit modeling of human-object interactions in realistic videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013.

[33] Mamshad Nayeem Rizve, Ugur Demir, Praveen Tirupattur, Aayush Jung Rana, Kevin Duarte, Ishan Dave, Yogesh Singh Rawat, and Mubarak Shah. Gabriella: An online system for real-time activity detection in untrimmed security videos. In *IEEE International Conference on Pattern Recognition*, 2020.

[34] Suman Saha, Gurkirt Singh, Michael Sapienza, Philip H.S. Torr, and Fabio Cuzzolin. Deep learning for detecting multiple space-time action tubes in videos. In *British Machine Vision Conference*, 2016.

[35] Abraham. Savitzky and M. J. E. Golay. Smoothing and differentiation of data by simplified least squares procedures. *Anal Chem*, 36(8):1627–1639, 1964.

[36] Yasaman S. Sefidgar, Arash Vahdat, Stephen Se, and Greg Mori. Discriminative key-component models for interaction detection and recognition. *Computer Vision and Image Understanding*, 135, 2015.

[37] Gurkirt Singh, Suman Saha, Michael Sapienza, Philip H.S. Torr, and Fabio Cuzzolin. Online real-time multiple spatiotemporal action localisation and prediction. *IEEE International Conference on Computer Vision (ICCV), 2017*, 2017.

[38] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *CoRR*, abs/1212.0402, 2012.

[39] Chris Stauffer and W. Eric L. Grimson. Learning patterns of activity using real-time tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22, 2000.

[40] Chen Sun, Abhinav Shrivastava, Carl Vondrick, Kevin Murphy, Rahul Sukthankar, and Cordelia Schmid. Actor-centric relation network. In *European Conference on Computer Vision*, pages 318–334, 2018.

[41] Namrata Vaswani, Amit Roy Chowdhury, and Rama Chellappa. Activity recognition using the dynamics of the configuration of interacting objects. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, 2003.

[42] Xiaolong Wang and Abhinav Gupta. Videos as space-time region graphs. In *European Conference on Computer Vision*, pages 413–431, 2018.

[43] Xiaoyang Wang and Qiang Ji. Video event recognition with deep hierarchical context model. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[44] Zhongdao Wang, Liang Zheng, Yixuan Liu, and Shengjin Wang. Towards real-time multi-object tracking. *CoRR*, 2019.

[45] Jianchao Wu, Zhanghui Kuang, Limin Wang, Wayne Zhang, and Gangshan Wu. Context-aware RCNN: A baseline for action detection in videos. In *European Conference on Computer Vision*, pages 440–456, 2020.

[46] Huijuan Xu, Abir Das, and Kate Saenko. R-C3D: region convolutional 3d network for temporal activity detection. In *IEEE International Conference on Computer Vision*, pages 5794–5803. IEEE Computer Society, 2017.

[47] Lijun Yu, Yijun Qian, Wenhe Liu, and Alexander G. Hauptmann. CMU informedia at TRECVID 2020: Activity detection with dense spatio-temporal proposals. In *TREC Video Retrieval Evaluation, TRECVID, 2020*, 2020.

[48] Yubo Zhang, Pavel Tokmakov, Martial Hebert, and Cordelia Schmid. A structured model for action detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[49] Yingying Zhu, Nandita M. Nayak, and Amit K. Roy-Chowdhury. Context-aware modeling and recognition of activities in video. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[50] Yingying Zhu, Nandita M. Nayak, and Amit K. Roy-Chowdhury. Context-aware activity modeling using hierarchical conditional random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37, 2015.