Draft of TRECVID2019 ActEV Evaluation Plan

(https://actev.nist.gov/trecvid19)

Date: 2019-08-12

ActEV Team, NIST

Contact: actev-nist@nist.gov

Table of Contents

1. Overview

The Activities in Extended Video (ActEV) series of evaluations is designed to accelerate development of robust, multi-camera, automatic activity detection systems for forensic and real-time alerting applications. ActEV began with the Summer 2018 Blind and Leaderboard evaluations and has currently progressed to the running of two concurrent evaluations: 1) the TRECVID 2019 ActEV self-reported leaderboard based on the VIRAT V1 and V2 datasets [7] with 18 target activities and, 2) an independent evaluation called the ActEV Sequestered Data Leaderboard (ActEV SDL) based on the MEVA dataset with 37 target activities.

The ActEV evaluation provides a mechanism for evaluating activity detection algorithms on challenging extended duration video. Activities in extended video are dispersed temporally and spatially requiring algorithms to detect and localize activities under a variety of collection conditions. Multiple activities may occur simultaneously in the same scene, while extended periods may contain no activities.

The TRECVID 2019 ActEV evaluation will be a leaderboard evaluation and will be run as an open activity detection evaluation where participants will run their algorithms on provided videos on their own hardware and submit results to the challenge scoring server of the National Institutes of Standards and Technology (NIST). The VIRAT V1 and V2 dataset will be used for the ActEV leaderboard evaluation.

For this evaluation plan, an activity is defined as "one or more people performing a specified movement or interacting with an object or group of objects". Detailed activity definitions are in the ActEV Annotation Definitions for VIRAT V1 and V2 [Appendix E]. Each activity is formally defined by four elements:

Element	**Meaning**	**Example Definition**
Activity Name	A mnemonic handle for the activity	Open Trunk
Activity Description	Textual description of the activity	A person opening a trunk
Begin time rule definition	The specification of what determines the beginning time of the activity	The activity begins when the trunk lid starts to move
End time rule definition	The specification of what determines the ending time of the activity	The activity ends when the trunk lid has stopped moving

Participants are also invited to submit their runnable activity detection software using an ActEV Command Line Interface (CLI) submission. NIST will then evaluate system performance on sequestered data using NIST hardware and results will be posted to a public leaderboard. The ActEV SDL participants will develop activity detection and temporal localization algorithms for 37 activities that are to be found in extended videos and video streams. These videos contain significant spans without any activities and intervals with potentially multiple concurrent activities. The ActEV SDL evaluation is based on the Multiview Extended Video with Activities (MEVA) dataset. Participants are encouraged to annotate the data for the 37 activities as described at mevadata.org. For ActEV participants, the MEVA dataset is available for download without a fee. For further information on evaluation plan and how to download the data, please refer the ActEV SDL website (actev.nist.gov/sdl).

2. Tasks and Conditions

2.1. Tasks

In the ActEV evaluation, there is one Activity Detection (AD) task for detecting and localizing activities.

For the AD task, given a target activity, a system automatically detects and temporally localizes all instances of the activity. For a system-identified activity instance to be evaluated as correct, the type of activity must be correct, and the temporal overlap must fall within a minimal requirement as described in Section 6.

2.2. Conditions

The ActEV evaluation will focus on the forensic analysis that processes the full corpus prior to returning a list of detected activity instances.

2.3. Evaluation Type

For open leaderboard evaluation, the challenge participants should run their software on their systems and configurations and submit the system output defined by this document (see Section 5) to the TRECVID 2019 ActEV Scoring Server (https://actev.nist.gov/trecvid19).

2.4. Protocol and Rules

The performers can train their systems or tune parameters using any data complying with applicable laws and regulations. All data used for training is expected to be made available by performers after the initial evaluation cycle where the data is used. In the event that external limitations preclude sharing such data with others, performers are still permitted to use the data, but they must inform NIST that they are using such data, and provide appropriate detail regarding the type of data used and the limitations on distribution.
The performers agree not to probe the test videos via manual/human means such as looking at the videos to produce the activity type and timing information from prior to the evaluation period until permitted by NIST.
All machine learning or statistical analysis algorithms must complete training, model selection, and tuning prior to running on the test data. This rule does not preclude online learning/adaptation during test data processing so long as the adaptation information is not reused for subsequent runs of the evaluation collection.
The only VIRAT data that may be used by the systems are the ActEV-provided training and validation sets, associated annotations, and any derivatives of those sets (e.g., additional annotations on those videos). All other VIRAT data and associated annotations may not be used by any of the systems for the ActEV evaluations.
For the reference temporal segmentation evaluation (when applicable), the performer must, to the extent possible, use the same underlying classifier for the evaluation. The provided segmentations are allowed to use for online learning/adaptation during test data processing.

2.5. Required Evaluation Condition

For TRECVID 2019 ActEV Leaderboard evaluation, the conditions can be summarized as shown in Table below:

ActEV 2019 Evaluation	**Required**
Task	Activity Detection
Target Application	Forensic Systems
Evaluation Type	Self-reported Leaderboard Evaluation
Submission	Primary (see the details in Appendix A for Submission Instructions)
Data Sets	VIRAT-V1 VIRAT-V2

3. Data Resources

The data used for TRECVID 2019 ActEV Leaderboard evaluation is the VIRAT V1 and V2 datasets (please see Appendix C to download data). The table below provides a list of 18

activities that are used in the ActEV evaluation. The definition of the 18 activities is given in Appendix D.

Closing
Closing_trunk
Entering
Exiting
Loading
Open_Trunk
Opening
Transport_HeavyCarry
Unloading
Vehicle_turning_left
Vehicle_turning_right
Vehicle_u_turn
Pull
Riding
Talking
activity_carrying
specialized_talking_phone
specialized_texting_phone

4. System Input

Along with the source video files, the subset of video files to process for evaluation will be specified in a provided file index JSON file. Systems will also be provided an activity index JSON file, which lists the activities to be detected by the system.

4.1. File Index

The file index JSON file lists the video source files to be processed by the system. Note that systems need only process the selected frames (as specified by the "selected" property). An example, along with an explanation of the fields is included below.

```
{
  "VIRAT_S_000000.mp4": {
      "framerate": 30,
      "selected": {
        "1": 1,
        "20941": 0
      }
  },
  "VIRAT_S_000001.mp4": {
```

```
      "framerate": 30,
      "selected": {
      "11": 1,
        "201": 0,
        "300": 1,
        "20656": 0
      }
  }
}
```

- <file>:
    - framerate: number of frames per second of video
    - selected: The on/off signal designating the evaluated portion of <file>
        - <framenumber>: 1 or 0, indicating whether or not the system will be evaluated for the given frame. Note that records are only added here when the value changes. For example in the above sample, frames 1 through 20940 in file "VIRAT_S_000000.mp4" are selected for processing/scoring. The default signal value is 0 (not-selected), and the frame index begins at 1, so for file "VIRAT_S_000001.mp4", frames 1 through 10 are not selected. Also note that the signal must be turned off at some point after it's been turned on, as the duration of the signal is needed for scoring.

4.2. Activity Index

The activity index JSON file lists the activities to be detected by the system. An example, along with an explanation of the fields is included below.

```
{
 "Closing": {},
 "Closing_Trunk": {},
 "Entering": {},
 "Exiting": {},
 "Loading": {}
}
```

- <activity>: A collection of properties for the given <activity>.

5. System Output

In this section, the system output format is defined. The ActEV Scorer software package[1] contains a submission checker that validates the submission in both the syntactic and semantic levels. Challenge participants should ensure their system output is valid because NIST will reject mal-formed output.

5.1. System Output File for Activity Detection Tasks

The system output file should be a JSON file that includes a list of videos processed by the system, along with a collection of activity instance records with spatio-temporal localization information (depending on the task). A notional system output file is included inline below, followed by a description of each field.

```
{
  "filesProcessed": [
    "VIRAT_S_000000.mp4"
  ],
  "activities": [
    {
      "activity": "Talking",
      "activityID": 1,
      "presenceConf": 0.89,
      "localization": {
        "VIRAT_S_000000.mp4": {
          "1": 1,
          "20": 0
        }
      }
    }
  ]
}
```

- filesProcessed: the list of video source files processed by the system
- activities: the list of detected activities; each detected activity is a record with the following fields:
    - activity: (e.g. “Talking”)
    - activityID: a unique identifier for the activity detection, should be unique within the list of activity detections for all video source files processed (i.e. within a single system output JSON file)
    - presenceConf: The score is any real number that indicates the strength of the possibility (e.g., confidence) that the activity instance has been identified. The scale of the presence confidence score is arbitrary but should be

[1]ActEV_Scorer software package (https://github.com/usnistgov/ActEV_Scorer)

consistent across all testing trials, with larger values indicating greater chance that the instance has been detected. Those scores are used to generate the detection error tradeoff (DET) curve.

- localization (temporal): The temporal localization of the detected activity for each file
  - <file>: The on/off signal temporally localizing the activity detection within the given <file>
    - <framenumber>: 1 or 0, indicating whether the activity is present or not, respectively. Systems only need to report when the signal changes (not necessarily every frame)

5.2. Validation of Activity Detection System Output

The system output file will be validated against a JSON Schema (see, ActEV Scorer: https://github.com/usnistgov/ActEV_Scorer). Further semantic checks may be performed prior to scoring by the scoring software. E.g. checking that the video list provided in the system output is congruent with the list of files provided to the system for evaluation.

To use the ActEV_Scorer to validate system output "SYSTEM.json", execute the following command:

```
% ActEV_Scorer.py ActEV19_AD_V2 -V -s SYSTEM.json  -a activity-index.json -f
file-index.json
```

6. Activity Detection Metrics

The technologies sought for the TRECVID 2019 ActEV leaderboard evaluation are expected to report activities that visibly occur in a single-camera video by identifying the video file, the frame span of the activity, and the *presenceConf* value indicating the system's 'confidence score' that the activity is present.

The primary measure of performance will be the normalized, partial Area Under the DET Curve ($nAUDC$) from 0 to a fixed, Time-based False Alarm (T_{fa}) value *a*, denoted $nAUDC_a$.

The partial area under DET curve is computed separately for each activity over all videos in the test collection and then is normalized to the range [0, 1] by dividing by the maximum partial area a. $nAUDC_a = 0$ is a perfect score. The $nAUDC_a$ is defined as:

$$nAUDC_a = \frac{1}{a} \int_{x=0}^{a} P_{miss}(x)\, dx, \quad x = T_{fa} \qquad (1)$$

where x is integrated over the set of T_{fa} values. T_{fa} and P_{miss} are defined as follows:

$$T_{fa} = \frac{1}{NR} \sum_{i=1}^{N_{frames}} max(0,\ S'_i - R'_i) \quad (2)$$

$$P_{miss}(x) = \frac{N_{md}(x)}{N_{TrueInstance}} \quad (3)$$

N_{frames} : The duration (frame-based) of the video
NR : Non-Reference duration. The duration of the video without the target activity occurring
S'_i : the total count of system instances for frame i
R'_i : the total count of reference instances for frame i
T_{fa} : The time-based false alarm value(see Section 6.1 for additional details)
$N_{md}(x)$: the number of missed detections at the presenceConf threshold that result in $T_{fa} = x$
$N_{TrueInstance}$: the number of true instances in the sequence of reference
$P_{miss}(x)$: The probability of missed detections (instance-based) value for $T_{fa} = x$ value (see Section 6.2 for additional details)

Implementation notes:

- If T_{fa} never reaches *a*, the system's minimum value of P_{miss} is used through a
- If the T_{fa} value occurs between two *presenceConf* values, a linearly interpolated value for *presenceConf* is used

6.1. Computation of Time-Based False Alarm

Time-based false alarm (T_{fa}) is the fraction of non-activity instance time (in the reference) for which there is a system that falsely identified an instance. All system instances, regardless of overlap with references instances, are included in this calculation and overlapping system instances contribute double or more (if there are more than two) to the false alarm time. Also note, temporally fragmented system detections that occur during non-activity time do not increase T_{fa} unless they overlap temporally.

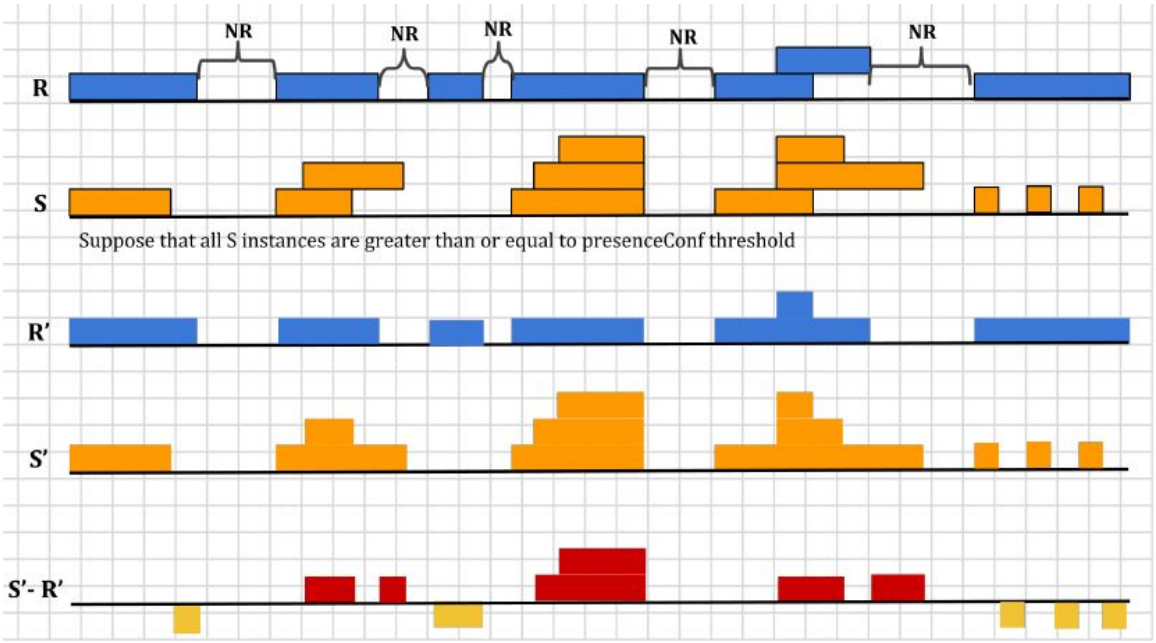

Figure 1: Pictorial depiction of T_{fa} calculation
(R is the reference instances and S is the system instances. R' is the histogram of the count of reference instances and S' is the histogram of the count of system instances for the target activity.)

In Equation (2), first the non-reference duration (*NR*) of the video where no target activities occurs is computed by constructing a time signal composed of the complement of the union of the reference instances durations. As depicted in the Figure above, R' and S' are histograms of count instances across frames (N_{frames}) for the reference instances (R) and system instances (S), respectively. R' and S' both have N_{frames} bins, thus R'_i is the value of the i^{th} bin of R' and S'_i is the value of the i^{th} bin of S'. S'_i is the total count of system instances in frame i and R'_i is the total count of reference instances in frame i.

False alarm time is computed by summing over positive difference of $S'_i - R'_i$ (shown in red in the figure above); that is the duration of falsely detected system instances. This value is normalized by the non-reference duration of the video to provide the T_{fa} value in Equation (2).

6.2. Alignment used in Computation of Probability of Missed Detection

A missed detection is a reference activity instance that the system did not detect. The Probability of Missed Detection (P_{miss}) is the fraction of reference instances not detected by the system.

As an instance-measure of performance, a single system instance cannot be counted as correct for multiple reference instances[2]. In order to optimally determine which instances are missed, and thereby minimize the measured P_{miss} , the evaluation code performs a reference-to-system instance alignment algorithm that minimizes the measured P_{miss} factoring the *presenceConf* values so that a single alignment also minimizes the $nAUDC$.

While the mapping procedure is one-to-one, system instances not mapped are ignored, effectively allowing a 1-to-many alignment because many system instances that overlap with a reference instance are not penalized in the P_{miss} calculation. However, all system instances can contribute to the T_{fa} calculation.

The alignment is computed between the reference instances and system detected instances using the Hungarian algorithm to the Bipartite Graph matching problem [2], which reduces the computational complexity and arrives at an optimal solution such that:

1. Correctly detected activity instances must meet a minimum temporal overlap with a single reference instance.
2. System instances can only account for one reference instance (otherwise, a single, full video duration system instance would be aligned to N reference instances).
3. The alignment prefers aligning higher presenceConf detections to minimize the measured error.

In bipartite graph matching approach, the reference instances are represented as one set of nodes and the system output instances are represented as one set of nodes. The mapping kernel function K below assumes that the one-to-one correspondence procedure for instances is performed for a single target activity (A_i) at a time.

$K(I_{R_i}, \varnothing) = 0$: the kernel value for an unmapped reference instance

$K(\varnothing, I_{S_j}) = -1$: the kernel value for an unmapped system instance

$$K(I_{R_i}, I_{S_j}) = \{\varnothing \text{ if } Activity(I_{S_j}) != Activity(I_{R_i})$$
$$\text{when } I_{S_j} >= 1 \text{ sec, } \varnothing \text{ if } Intersection(I_{R_i}, I_{S_j}) < 1 \text{ sec,}$$
$$\text{when } I_{S_j} < 1 \text{ sec, } \varnothing \text{ if } Intersection(I_{R_i}, I_{S_j}) < 50\% \text{ of } I_{R_i} \text{ time}$$
$$1 + AP_{con}(I_{S_j}), \quad otherwise\}$$

where,

[2] For instance, if there are two abandon_bag activity instances that occur at the same time but in separate regions of the video and there was a single detection by the system, one of the reference instances was missed.

$$AP_{con}(I_{s_j}) = \frac{AP(I_{s_j}) - AP_{min}(S_{AP})}{AP_{max}(S_{AP}) - AP_{min}(S_{AP})}$$

A_i : the activity label of an instance
I_{R_i} : the i^{th} reference instance of the target activity
I_{S_j} : the j^{th} system output instance of the target activity
K : the kernel score for activity instance I_{R_i} , I_{S_j}
$Intersection(I_{R_i}, I_{S_j})$: the time span intersection of the instances I_{R_i} , I_{S_j}
$AP_{con}(I_{S_j})$: a presence confidence score congruence of system output activity instances
$AP(I_{S_j})$: the presence confidence score of activity instance I_{S_j}
S_{AP} : the system activity instance presence confidence scores that indicates the confidence that the instance is present
$AP_{min}(S_{AP})$: the minimum presence confidence score from a set of presence confidence scores, S_{AP}
$AP_{max}(S_{AP})$: the maximum presence confidence score from a set of presence confidence scores, S_{AP}

$K(I_{R_i}, I_{S_j})$ has the two values; $\varnothing$ indicates that the pairs of reference and system output instances are not mappable due to either missed detections or false alarms, otherwise the pairs of instances have a score for potential match.

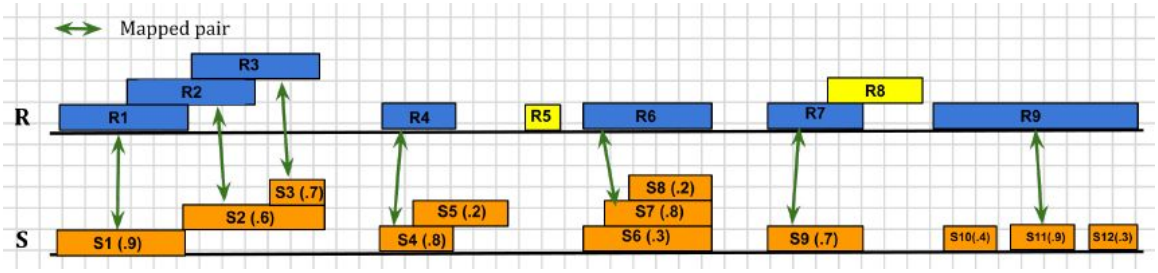

Figure 2: Pictorial depiction of activity instance alignment and P_{mss} calculation
(In *S*, the first number indicates instance id and the second indicates *presenceConf* score. For example, S1 (.9) represents the instance S1 with corresponding confidence score 0.9. Green arrows indicate aligned instances between R and S.)

In the example of Figure 2, for the case of reference instances {R1, R2, R3} and system instances {S1, S2, S3}, either R2 or R3 can be considered as a missed detection depending on the way reference instances are mapped to system instances. To minimize P_{miss} for such cases, the alignment algorithm is used to determine one-to-one correspondence as to {R1, S1}, {R2, S2}, and {R3, S3}. It also identifies system instance S7 as a better match to reference instance R6 factoring the *presenceConf* values.

In Equation (3), $N_{TrueInstance}$ represents the number of true instances in the sequence of reference and N_{md} is the number of nonaligned reference instances that are missed by the system. In Figure 2, suppose that the *presenceConf* threshold is greater than or equal to 0.5. Thereby, $N_{TrueInstance}$ is 9 and N_{md} is 2 (marked in yellow).

6.3. ACTEV_Scoring Command Line

The command to score a system using the ActEV_Scorer is:

```
% ActEV_Scorer.py ActEV19_AD -s system-output.json -r reference.json -a
    activity-index.json -f file-index.json -o output-folder -F -v
```

The command to validate system-generated output using the ActEV_Scorer is:

```
% ActEV_Scorer.py ActEV19_AD -s system-output.json -a activity-index.json -f
    file-index.json -F -v -V
```

APPENDIX

Appendix A: Submission Instructions

System output and documentation submission to NIST for subsequent scoring must be made using the protocol, consisting of three steps: (1) preparing a system description and self-validating system outputs, (2) packaging system outputs and system descriptions, and (3) transmitting the data to NIST.

The packaging and file naming conventions for the TRECVID 2019 ActEV rely on **Submission Identifiers** (SubID) to organize and identify the system output files and system description for each evaluation task/condition. Since SubIDs may be used in multiple contexts, some fields contain default values. The following EBNF (Extended Backus-Naur Form) describes the SubID structure with several elements:

<SubID> ::= <SYS>_<VERSION>_[OPTIONAL]

<SYS> is the SysID or system ID. No underscores are allowed in the system ID. The team is allows to have the two submissions only; primary and secondary respectively. It should begin with 'p-' for the one primary system (i.e., your best system) or with 's-' for the one secondary system. It should then be followed by an identifier for the system (only alphanumeric characters allowed, no spaces). For example, this string could be "p-baseline" or "s-deepSpatioTemporal". This field is intended to differentiate between runs for the same evaluation condition. Therefore, a different SysID should be used for runs where any changes were made to a system.

<VERSION> should be an integer starting at 1, with values greater than 1 indicating multiple runs of the same experiment/system.

[OPTIONAL] is any additional strings that may be desired, e.g. to differentiate between tasks. This will not be used by NIST and is not required. If left blank, the underscore after <VERSION> should be omitted.

As an example, if the team is submitting on the AD task using their third version of the primary baseline system, the SubID could be:

p-baseline_3_AD

A-a System Descriptions

Documenting each system is vital to interpreting evaluation results. As such, each submitted system, determined by unique experiment identifiers, must be accompanied by a system description with the following information.

Section 1 Submission Identifier(s)

List all the submission IDs for which system outputs were submitted. Submission IDs are described in further detail above.

Section 2 System Description

A brief technical description of your system.

Section 3 System Hardware Description and Runtime Computation

Describe the computing hardware setup(s) and report the number of CPU and GPU cores. A hardware setup is the aggregate of all computational components used.

Report salient runtime statistics including: wall clock time to process the index file, resident memory size of the index, etc.

Section 4 Speed Measures and Requirements

For the TRECVID 2019 ActEV evaluation the challenge participants will report the processing speed per video stream compared to real-time by running only on one node of their system for each task separately. For this challenge, real-time processing refers to processing at the same rate as the input video.

For the TRECVID 2019 ActEV Leaderboard evaluation the challenge participants will report the processing speed per video stream compared to real-time by running only on one node of their system for the AD task.

Section 5 Training Data and Knowledge Sources

List the resources used for system development and runtime knowledge sources beyond the provided ActEV dataset.

Section 6 System References

List pertinent references, if any.

A-b Packaging Submissions

Using the SubID, all system output submissions must be formatted according to the following directory structure:

<SubID>/

<SubID>.txt The system information file, described in Appendix A-a

<SubID>.json The system output file, described in Section 5.1

As an example, if the earlier team is submitting, their directory would be:

p-baseline_3_AD/

p-baseline_3_AD.txt

p-baseline_3_AD.json

A-c Transmitting Submissions

To prepare your submissions, first create the previously described file/directory structure. Then, use the command-line example to make a compress the TAR or ZIP file:

$ tar -zcvf SubID.tgz SubID/ e.g., tar -zcvf p-baseline_3_AD.tgz p-baseline_3_AD/

$ zip -r SubID.zip SubID/ e.g., zip -r p-baseline_3_AD.zip p-baseline_3_AD/

Please submit your files in time for us to deal with any transmission errors that might occur well before the due date if possible. Note that submissions received after the stated due dates for any reason will be marked late.

Appendix B: JSON Schemas for System Output File

Please refer to the ActEV_Scorer software package (same for the ActEV evaluations) (https://github.com/usnistgov/ActEV_Scorer) for the most up-to-date schemas, found in "lib/protocols".

Appendix C: Infrastructure (Hardware and Virtual Machine specification)

Scoring Server

The team will submit their system output in the Json file format described earlier to an online web based evaluation server application at NIST. The initial creator of the team on the scoring server will have control over who can submit system outputs on behalf of the team using a username and a password. The evaluation server will validate the file format and then compute scores. The scores will be manually reviewed by the DIVA T&E team prior to dissemination. The server will be available for teams to test the submission process.

Appendix D: Data Download

To download the data, complete these steps:

Get an up-to-date copy of the ActEV Data Repo via GIT. You'll need to either clone the repo (the first time you access it) or updated a previously downloaded repo with 'git pull'. Note: this is the same repo as used for MEVA.

Clone: git clone https://gitlab.kitware.com/actev/actev-data-repo.git

Update: cd "Your_Directory_For_actev-data-repo"; git pull

Add VIRAT-V1 and VIRAT-V2 download credentials:

Change your working directory the top-level of the repo.

cd "Your_Directory_For_actev-data-repo"

Follow the steps in the top-level README.

For Step 2 in the download instructions, use these two commands to add your access credentials. (Please do not email this command!)

python ./scripts/actev-corpora-maint.py --operation summary --corpus VIRAT-V1 --add_credential '{"corpus": "VIRAT-V1", "urls": {"https://mig.nist.gov/datasets/VIRAT-V1": {"type": "file_store", "user": "VIRATv1", "password": "??????"}}}'

python ./scripts/actev-corpora-maint.py --operation summary --corpus VIRAT-V2 --add_credential '{"corpus": "VIRAT-V2", "urls": {"https://mig.nist.gov/datasets/VIRAT-V2": {"type": "file_store", "user": "VIRATv2", "password": "??????"}}}

Appendix E: Definitions of Activity and Associated objects [6]

For the ActEV 2019 evaluations, the definitions of the 18 target activity and the objects associated with the activity are described below [6].

Closing

Closing Description: A person closing the door to a vehicle or facility.
Start: The event begins 1 s before the door starts to move.
End: The event ends after the door stops moving. People in cars who close the car door from within is a closing event if you can still see the person within the car. If the person is not

visible once they are in the car, then the closing should not be annotated as an event.
Objects associated with the activity : Person; and Door or Vehicle

Closing_trunk

Close Trunk Description: A person closing a trunk. See Open Trunk (above) for definition of trunk and special cases.
Start: The event begins 1 s before the trunk starts to move.
End: The event ends after the trunk has stopped moving.
Objects associated with the activity: Person; and Vehicle

Entering

Entering Description: A person entering (going into or getting into) a vehicle or facility.
Start: The event begins 1 s before the door moves or if there is no door, the event begins 1 s before the person's body is inside the vehicle/facility.
End: The event ends when the person is in the vehicle/facility and the door (if present) is shut.
Notes: A facility is defined as a structure built, installed or established to serve a particular purpose. This facility must have an object track (e.g., door or doorway) for the person to enter through. The two necessary tracks included in this event are
(1) the person entering and (2) the vehicle or the object for entering a facility (e.g., door). A special case of "entering" is mounting a motorized vehicle (e.g., motorcycle, powered scooter).
Note 2 : No special activity for standing or crouching when entering or exiting a vehicle. Whenever the person starts standing or walking, annotate as usual, but once they stop lateral motion and start bending down to get into out of the car, they've stopped both standing and walking, so no activity. Sitting in car when entering or exiting is only if sitting is visible for >10 frames.
Objects associated with the activity: Person; and Door or Vehicle

Exiting

Exiting Description: A person exiting a vehicle or facility. See entering for definition of facility.
Start: The event begins 1 s before the door moves or if there is no door, the event begins 1 s before half of the person's body is outside the vehicle/facility.
End: The event ends 1 s after the person has exited the vehicle/facility.
Note: A special case of "exiting" is dismounting a motorized vehicle (e.g., motorcycle, motorized scooter).
Objects associated with the activity: Person; and Door or Vehicle

Loading

Loading Description: An object moving from person to vehicle.
Start: The event begins 2 s before the cargo to be loaded is extended toward the vehicle (i.e., before a person's posture changes from one of "carrying" to one of "loading").
End: The event ends after the cargo is placed into the vehicle and the person-cargo contact is lost. In the event of occlusion, it ends when the loss of contact is visible.
Note: The two necessary tracks included in this event are the person performing the (un)loading and the vehicle/cart being (un)loaded. Additionally, if the items being loaded are at least half the person's size or large enough to substantially modify the person's gait (as defined in the Carrying activity -- 4.7), then they should be
individually tracked as Props and included in the event. "Fiddling" with the object being (un)loaded is still part of the (un)loading process.
Objects associated with the activity: Person; and Vehicle

Open_Trunk

Open Trunk Description: A person opening a trunk. A trunk is defined as a container designed to store non-human cargo on a vehicle.
Start: The event begins 1 s before the trunk starts to move.
End: The event ends after the trunk has stopped moving.
Notes: A trunk does not need to have a lid or open from above. So the back/bed of a truck is a trunk and dropping the tailgate is the equivalent of opening a trunk. Additionally, opening the double doors on the back of a van is the equivalent of opening a trunk.
Objects associated with the activity: Person; and Vehicle

Opening

Opening Description: A person opening the door to a vehicle or facility.
Start: The event begins 1 s before the door starts to move.
End: The event ends after the door stops moving.
Note: The two necessary tracks included in this event are (1) the person opening the door and (2) the vehicle or the object for a facility (e.g., door). The vehicle door does not need to be independently annotated because the vehicle itself is a track which can be coupled to the person in this event. This event often overlaps with entering/exiting; however, can be independent or absent from these events.
Note 2: Opening clarification: When opening a car door, the event ends when the when the door stops moving from being opened. This is distinguished from someone opening a car door, then leaning on the door when they exit and the door wiggles.
The wiggling is not part of opening, even though it is in fact moving.
Objects associated with the activity : Person; and Door or Vehicle

Transport_HeavyCarry

Transport Large Object or Heavy Carry Description: A person or multiple people carrying an oversized or heavy object. This is characterized by the object being large enough (over half the size of the person) or heavy enough (where the person's gait has been substantially modified) to require being tracked separately.
Start: This event begins 1 s before the person (or the first person for multiple people) establishes contact with the object.
End: This event ends 1 s after the person (or the final person for multiple people) loses contact with the object.
Objects required : Person; and Prop

Unloading

Unloading Description: An object moving from vehicle to person.
Start: The event begins 2 s before the cargo begins to move. If the start of the event is occluded, then it begins when the cargo movement is first visible.
End: The event ends after the cargo is released. If the person holding the cargo begins to walk away from the vehicle, the event ends after 1 s of walking. If the door is closed on the vehicle, the event ends when the door is closed. If both of these things happen, the event ends at the earlier of the two events.
Note: See Loading above.
Objects associated with the activity: Person; and Vehicle

Vehicle_turning_left

Turning Left Description: A vehicle turning left or right is determined from the POV of the driver of the vehicle. The vehicle may not stop for more than 10 s during the turn.
Start: Annotation begins 1 s before vehicle has noticeably changed direction.
End: Annotation ends 1 s after the vehicle is no longer changing direction and linear motion has resumed.
Note: This event is determined after a reasonable interpretation of the video.
Objects associated with the activity : Vehicle

Vehicle_turning_right

Turning Right Description: A vehicle turning left or right is determined from the POV of the driver of the vehicle. The vehicle may not stop for more than 10 s during the turn.
Start: Annotation begins 1 s before vehicle has noticeably changed direction.
End: Annotation ends 1 s after the vehicle is no longer changing direction and linear motion has resumed.
Note: This event is determined after a reasonable interpretation of the video.
Objects associated with the activity : Vehicle

Vehicle_u_turn

U-Turn Description: A vehicle making a u-turn is defined as a turn of 180 and should give the appearance of a “U”. A u-turn can be continuous or comprised of discrete events (e.g., a 3-point turn).The vehicle may not stop for more than 10 s during the u-turn.
Start: Annotation begins when the vehicle has ceased linear motion.
End: Annotation ends 1 s after the car has completed u-turn.
Note: This event is determined after a reasonable interpretation of the video. U-turns do not contain left and right turns (or start/stop in the case of K turns). U-turns are also annotated when going around something, like a bank of trees/shrubs.
Objects associated with the activity: Vehicle

Pull

Pull Description: A person exerting a force to cause motion toward. The two necessary tracks included in this event are the person pulling and object being pulled (Push/Pulled Object - See Active Object Type 3.5).
Start: As soon as the object is visibly moving or track begins if object already in motion.
End: As soon as the object is no longer moving or the person loses contact with the object being pulled. In the event of occlusion, the event will end when the loss of contact is visible.
Objects required : Person; and Push/Pulled Object

Riding

Riding Description: A person riding a “bike” (i.e., any one of the variety of human powered vehicles where the person is still visible but their movement is modified).
Note: The two necessary tracks included in this event are (1) the person and (2) the “bike” they are riding. Events for Riding, Pushing and Pulling are used to couple the person and “bike” tracks.
Start: This event begins when the person’s motion is modified by the “bike”, or upon entering the FOV if the person is already riding the “bike”.
End: This event ends when the person’s motion is no longer modified by the “bike”, or upon exiting the FOV Objects associated with the activity: Person(s);

Talking

Talking Description: A person talking to another person in a face-to-face arrangement between n + 1 people.
Start: This event begins when the face-to-face arrangement is initiated.
End: This event ends when the face-to-face arrangement is broken.
Objects associated with the activity: Person(s);

Activity_carrying

Carrying Description: A person carrying an object up to half the size of the person, where the person's gait has not been substantially modified. The object may be carried in either

hand, with both hands, or on one's back.
Examples: Carrying a Backpack, Purse, Briefcase, or Box.
Counter-examples: "Incidental carrying" such as a sheet of paper or a file folder such that the person's arm motion is not affected by the payload.
Start: Annotation begins in one of two ways: (1) when the person who will be carrying the object makes contact with the object, or (2) when the track begins, if the person is already carrying the object (e.g., backpack or purse).
End: Annotation ends when contact with the object is broken.
Note: If a carried object (e.g., purse, backpack, box) is separated from the individual, a new track for that object ("Prop") will be created. The events, pickup, drop, and set down will be used to couple/decouple the person and object.
Objects associated with the activity: Person(s);

Specialized_talking_phone

Talking On Phone Description: A person talking on a cell phone where the phone is being held on the side of the head. This activity should apply to the motion of putting one's hand up to the side of their head regardless of the presence of a phone in hand.
Start: Annotation should begin when hand makes motion toward side of head.
End: Annotation should end 1 s after hand leaves side of head.
Objects associated with the activity: Person(s);

Specialized_texting_phone

Texting On Phone Description: A person texting on a cell phone. This applies to any situation when the phone is in front of the person's face (as opposed to along the side of the head) and they are using it. This includes using the phone with thumbs and fingers or video chatting.
Start: Annotation should begin 1 s before "texting" is observed.
End: Annotation should end 1 s after the last instance of "texting" is observed.
Objects associated with the activity: Person(s);

References

[1] TRECVID 2017 Evaluation for Surveillance Event Detection, https://www.nist.gov/itl/iad/mig/trecvid-2017-evaluation-surveillance-event-detection
[2] J. Munkres, "Algorithms for the assignment and transportation problems," Journal of the Society of Industrial and Applied Mathematics, vol. 5, no. 1, pp. 32–38, 1957
[3] Martin, A., Doddington, G., Kamm, T., Ordowski, M., Przybocki, M., "The DET Curve in Assessment of Detection Task Performance", Eurospeech 1997, pp 1895-1898.
[4] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: The clear mot metrics," EURASIP Journal on Image and Video Processing, vol. #, 2008.

[5] R.Kasturi et al.,"Framework for performance evaluation of face, text, and vehicle detection and tracking in video: Data, metrics, and protocol," IEEE Trans. Pattern Anal. Mach. Intell., vol. 31, no. 2, pp. 319–336, Feb. 2009.
[6] Kitware DIVA Annotation Guidelines, Version 1.0 November 6, 2017.
[7] VIRAT Video Dataset: http://www.viratdata.org/

Disclaimer

Certain commercial equipment, instruments, software, or materials are identified in this evaluation plan to specify the experimental procedure adequately. Such identification is not intended to imply recommendation or endorsement by NIST, nor is it intended to imply that the equipment, instruments, software or materials are necessarily the best available for the purpose.