Draft ActEV 2019 Evaluation Plan

Date: 2019-04-25

ActEV Team

NIST

Table of Contents

Background **4**

Overview **4**

2. Tasks and Conditions **5**

2.1. Tasks 5

2.1.1. Activity Detection (AD) 5

2.2. Conditions 5

2.3. Evaluation Type 5

2.3.1. TrecVID ActEV 2019 leaderboard evaluation 5

2.3.2. ActEV 2019 Independent evaluation 5

2.3. Evaluation Type 5

2.3.1. Self-reported evaluation 6

2.3.2. Independent/Sequestered evaluation 6

2.4. Protocol and Rules 6

2.5. Required Evaluation Condition 6

3. Data Resources **7**

4. System Input **8**

4.1. File Index 8

4.2. Activity Index 8

5. System Output **9**

5.1. System Output File for Activity Detection Tasks 9

5.2. Validation of Activity Detection System Output 10

6. Activity Detection Metrics **11**

6.1. Activity Detection Task 11

6.1.1. Activity Instance occurrence detection per activity (ActEV18_AD Protocol Primary Metric ActEV-PC) 11

6.1.2. Activity Instance occurrence detection per activity (ActEV18_AD_1SECOL) 15

6.1.3. Activity Instance occurrence detection per activity (ActEV18_AD_TFA) 15

6.1.4. Activity Instance occurrence detection per activity (ActEV19_AD Protocol Primary Metric for TrecVID 2019 ActEV Evaluation) 16

6.5. System Information 16

6.5.1. System Description 16

6.5.2. System Hardware Description and Runtime Computation 16

6.5.2.1. Speed Measures and Requirements 16

6.5.3. Training Data and Knowledge Sources 16

6.5.4. System References 16

APPENDIX 17

Appendix A: Submission Instructions 17

Appendix B: SCHEMAS 19

JSON Schema for system output file 19

Appendix C: Infrastructure (Hardware and Virtual Machine specification) 19

Scoring Server 19

NIST Independent Evaluation Infrastructure Specification 19

Independent Evaluation Infrastructure and Delivery of Software 20

Appendix C: Data Download 20

Appendix D: Definitions of Activity and Required objects [6] 21

References 25

Disclaimer 25

Background

The volume of video data collected from ground-based video cameras has grown dramatically in recent years. However, there has not been a commensurate increase in the usage of intelligent analytics for real-time alerting or triaging of video. Operators of camera networks are typically overwhelmed with the volume of video they must monitor, and cannot afford to view or analyze even a small fraction of their video footage. Automated methods that identify and localize activities in extended video are necessary to alleviate the current manual process of monitoring by human operators and provide the capability to alert and triage video that can scale with the growth of sensor proliferation.

Overview

The TrecVID 2019 Activities in Extended Video (ActEV 2019) evaluation seeks to encourage the development of robust automatic activity detection algorithms for a multi-camera streaming video environment. Challenge participants will develop activity detection and temporal localization algorithms for 18 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.

ActEV 2019 will be an 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 2019 leaderboard evaluation.

NIST will also run a ActEV Independent evaluation in 2019 based on the MEVA video data for the 38 activities (details coming soon). We would like the ActEV 2019 participants to help us annotate the MEVA training data for the 38 activities as defined in the annotation guide (coming soon).

For this evaluation plan, an activity is defined to be "one or more people performing a specified movement or interacting with an object or group of objects". Activities are determined during annotations and defined in the data selections below. 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

2. Tasks and Conditions

2.1. Tasks

In the TrecVID ActEV 2019 evaluation, there is one Activity Detection (AD) task for detecting and localizing of activities .

2.1.1. Activity Detection (AD)

For the Activity Detection task, given a target activity, a system automatically detects and temporally localized 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 2019 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 the ActEV 2019 evaluation, there will be two types of evaluation; a self-reported TrecVID ActEV 2019 leaderboard evaluation and an actEV 2019 independent evaluation for the selected participants.

2.3.1. TrecVID ActEV 2019 leaderboard evaluation

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 NIST ActEV Scoring Server (https://actev.nist.gov/).

2.3.2. ActEV 2019 Independent evaluation

The selected participants will provided their runnable system to NIST using the Evaluation Container Submission Instructions. NIST will evaluate system performance on sequestered data using NIST hardware--see the details in Appendix C for the hardware infrastructure.

2.3. Evaluation Type

For the ActEV evaluation, there are the two evaluation types; self-reported evaluation and sequestered evaluation.

2.3.1. Self-reported evaluation

For self-reported evaluation, the performers should run their software on their systems and configurations and submit the system output defined by this document (see Section 5) to the NIST Scoring Server.

2.3.2. Independent/Sequestered evaluation

For independent/sequestered evaluation, the performers should submit their runnable system to NIST using the forthcoming Evaluation Container Submission Instructions. NIST will evaluate system performance on sequestered data using NIST hardware--see the details in Appendix C for the hardware infrastructure.

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

ActEV 2019 Evaluation	**Required**
Task	AD
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

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

ActEV 2019 Independent Evaluation	**Required**
Task	AD
Target Application	Forensic Systems
Evaluation Type	Independent Evaluation
Submission	Primary (see the details in Appendix A for Submission Instructions)
Data Sets	VIRAT-V1 VIRAT-V2 MEVA

3. Data Resources

This data used for ActEV 2019 Leaderboard evaluation is the VIRAT V1 and V2 datasets and the ActEV 2019 Independent evaluation is based on sequestered MEVA dataset.

The table below provides a list of activities for the ActEV 2019 evaluations. The eighteen target activities are used in both the ActEV 2019 leaderboard and the ActEV 2019 independent evaluation will be based on 38 activities. The detailed definitions of the activities and its associated objects are found in Appendix D.

Table: List of activities per task and required objects

ActEV-PC evaluation (phase 1 and 2)	**ActEV 2019 Leaderboad evaluation**
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	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 with 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>
    - objectTypes: the set of objects to be detected by the system for the given activity

5. System Output

In this section, the types of system outputs are defined. The ActEV Score package[1] contains a submission checker that validates the submission in both the syntactic and semantic levels. Challenge participants should check their submission prior to sending them to NIST. We will reject submissions that do not pass validation. The ActEV Scoring Primer document contains instructions for how to use the validator. NIST will provide the command line tools to validate submission files.

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. Regarding file naming conventions for submission, please refer to Appendix A.

Note that some fields may be optional depending on which task the system output is submitted for.

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

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

```
          "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 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 localizating 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)

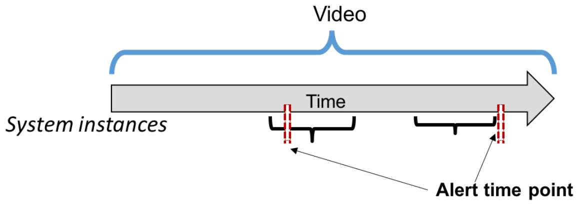

Figure 1. Example of system reported alert time

5.2. Validation of Activity Detection System Output

The system output file will be validated against a JSON Schema (see Appendix B), 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 teams for evaluation.

6. Activity Detection Metrics

The technologies sought for the ActEV 2019 leaderboard evaluation are expected to report activities that occur in the ensemble of video(s) by identifying the camera(s) for which the activity is visible, reporting the time span of the activity. The approach to evaluating system performance for the AD task. The primary measures, will compute system performance using the binary detection measure of the probability of missed detection (P_{miss}) at a time based false alarm (TFA). System performance on ActEV 2019 is measured by:

Tasks	Primary Question/Metric	Evaluated System Instance Content
AD	Can a system temporally detect instances of a target activity X? $P_{miss}@TFA = X$	Activity, BeginFrame, EndFrame, Score

The following description is a scoring protocol for the primary performance measure. Given reference and system output, in general, the scoring procedure can be divided into four distinctive steps: Alignment, Detection Confusion Matrix, and Performance Metrics.

6.1. Activity Detection Task

The ActEV18_AD protocol (described section 6.1.1.) was used during the ActEV-PC phase 1 and 2. For the TrecVID 2019 ActEV evaluation and other evaluations in 2019 we will use the ActEV19_AD protocol, which is described in more detail in the following sections 6.1.2., 6.1.3. and 6.1.4.

The changes from the user point of view is the addition of three new scoring protocols to use on the command line of the ActEV scorer (ActEV18_AD_TFA, ActEV18_AD_1SECOL, ActEV19_AD). The scoring protocols are summarized below.

6.1.1. Activity Instance occurrence detection per activity (**ActEV18_AD Protocol** Primary Metric ActEV-PC)

This metric evaluates performance on whether the system correctly detects the presence of the target activity instance. The measure requires a one-to-one correspondence between pairs of reference and system output activity instances. The following descriptions are a modified version of the TRECVID 2017: Surveillance Event Detection [1] framework.

Step 1: Instance Alignment (One-to-One Correspondence)

For a target activity, multiple instances can occur within the same duration. For example, instances R_2 and R_3 occur in different locations within the same duration in a video as shown in Figure 2.

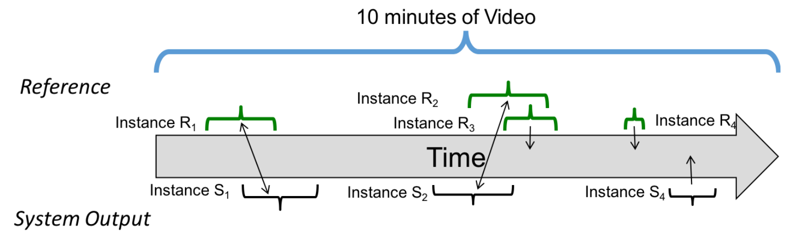

Figure 2. The instance alignment of the reference and system output

To compare the instances of the system output and the reference annotations, the scoring tool will first find the corresponding instances between reference and system output. For an optimal one-to-one instance mapping, the tool utilizes the Hungarian solution to the Bipartite Graph matching problem [2], which reduces the computational complexity and arrives at an optimal solution.

In this 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

$$K(I_{R_i}, I_{S_j}) = \{ \varnothing \; if \; Activity \; (I_{S_j}) \; ! = Activity \; (I_{R_i})$$

$$\varnothing \; if \; Temporal_{IoU}(I_{R_i}, I_{S_j}) <= \Delta Temporal_{IoU}$$

$$1 + E_{IoU} * Temporal_{IoU}(I_{R_i}, I_{S_j}) + E_{AP} * AP_c(I_{S_j}), \quad otherwise \}$$

where,

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

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}

$Union(I_{R_i}, I_{S_j})$: the time span union of the instances I_{R_i} , I_{S_j}

$Temporal_{IoU}(I_{R_i}, I_{S_j})$: $Intersection(I_{R_i}, I_{S_j})$ over $Union(I_{R_i}, I_{S_j})$ in a temporal domain

$\Delta Temporal_{IoU} = 0.2$; the fixed temporal Intersection over Union (IoU) threshold

$E_{IoU} = 0$; a constant to weight overlap ratio congruence

$E_{AP} = 1$; a constant to weight activity presence confidence score congruence

$AP_c\ (I_{S_j})$: 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.

The constants E_{IoU} and E_{AP} have two functions: first they set the relative importance of the information sources, (temporal IoU, activity presence confidence scores respectively). Second, they control the information source used to alignment. For example, if $E_{AP} = 0$ the presence confidence score has no bearing on the alignment and resulting performance scores.

Note that the kernel function is used to find the corresponding instance pair between reference and system output, not to measure accuracy of the system performance. The components, however, influence the performance metrics (e.g., P_{miss} and $Rate_{FA}$)--for example, an incorrect object detection can cause the system detected instance set to miss detections.

Step 2: Confusion Matrix Computation

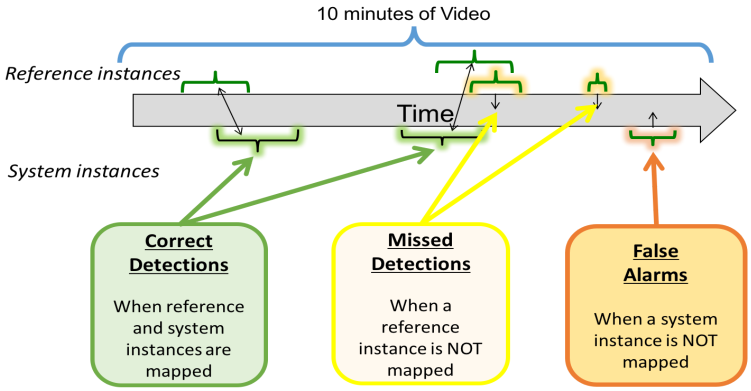
10 minutes of Video
Reference instances
Time
System instances
Correct Detections
When reference and system instances are mapped
Missed Detections
When a reference instance is NOT mapped
False Alarms
When a system instance is NOT mapped

Figure 3. Confusion matrix computation

Figure 3, illustrates the confusion matrix calculation for activity instance occurrence and the matrix is defined as:

- Correct Detection (CD): when reference and system output instances are mapped as a correct correspondence. The example instances are shown in green.
- Missed Detection (MD): when an instance in the reference has no correspondence to an instance with same label in the system output. The example instances are shown in yellow.
- False Alarm (FA): when an instance in the system output has no correspondence to an instance with same label in the reference. The example instances are shown in orange.
- Correct Rejection (CR): The reference indicates it is no instance for duration, and the system output also does not detect it as an instance. This is not computable in this evaluation.

Step 3: Performance Metrics

Following the calculation of the detection confusion matrix, the next step is to summarize the performance metrics. Each trial's score will be converted to a decision by comparing the score to a certain threshold; a trial with a score greater than or equal to the threshold is interpreted as a decision of "yes", indicating that the system's belief is that the activity instance is a target activity; a trial with a score less than the threshold is interpreted as a decision of "no", indicating that the system's belief is that the instance is not a target activity. For activity instance occurrence, a probability of missed detections and a rate of false alarms at a given threshold τ can be computed:

$$P_{miss}(\tau) = \frac{8+N_{MD}(\tau)}{10+N_{TrueInstance}}$$

$$Rate_{FA}(\tau) = \frac{N_{FA}(\tau)}{VideoDurationInMinutes}$$

$P_{miss}(\tau)$: the probability of missed detections at the activity presence confidence score threshold τ.

$Rate_{FA}(\tau)$: the rate of false alarms at the presence confidence score threshold τ.
$N_{MD}(\tau)$: the number of missed detections at the presence confidence score threshold τ.
$N_{FA}(\tau)$: the number of false alarms at the presence confidence score threshold τ.
$N_{TrueInstance}$: the number of true instances in the sequence

The P_{miss} is calculated as a weighted P_{miss}, and this is a regularizer that is most relevant for activities that have few instances. The regularizer incorporates our prior belief on PMiss being around .8, based on previous research. Activities for which there are many instances to detect will overcome this prior, and activities for which there are not many instances will be more weighted by the prior. This value is then averaged over all activities. Each activities' counts in the leaderboard and sequestered data will not be published, but activities' relative counts will differ from public datasets.

For ActEV-PC evaluations, system performance will be evaluated at the operating points; P_{miss} at $Rate_{FA}$=0.15 for activities.

The changes from the user point of view is the addition of three new scoring protocols to use on the command (ActEV18_AD_TFA,ActEV18_AD_1SECOL, ActEV19_AD). The scoring protocols are summarized below

6.1.2. Activity Instance occurrence detection per activity (**ActEV18_AD_1SECOL**)

This protocol relaxes temporal overlap of system/reference activity instances. They are required to overlap by greater than 1 second (rather than the intersection-over-union minimum in section 6.1.1 of the Eval plan). Further, the $Temporal_{IoU}(I_{R_i}, I_{S_j})$ term in the kernel function was removed.

There's a new threshold-based scoring file 'scores_by_activity_and_threshold.csv'

6.1.3. Activity Instance occurrence detection per activity (**ActEV18_AD_TFA**)

This protocol implements the new "Time-Based False Alarm" (TFA) measure for system evaluation.

TFA is defined to be the duration of system instances that do not overlap ANY reference instances divided by the time of video for which no reference annotations exist.

$$TFA=|NR - Sys_s|/|NR|$$

This protocol generates new measurements as follows:

New DET Curves: figures/DET_TFA_*

New TFA lines in the scores files with the label 'p_miss@0.03tfa'. We presently are using the same thresholds as rfa. We are going to revise these TFA thresholds after the program establishes new performance goals.

6.1.4. Activity Instance occurrence detection per activity (**ActEV19_AD Protocol** Primary Metric for TrecVID 2019 ActEV Evaluation)

This combines both the 1-Second time constraint (ActEV18_AD_1SECOL) and the new time-based false alarm metric (ActEV18_AD_TFA protocol). This is the primary metric for the TrecVID ActEV 2019 Evaluation.

6.5. System Information

6.5.1. System Description

A brief technical description of your system. Please see the detailed format in Appendix A-a System Descriptions.

6.5.2. 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.

6.5.2.1. Speed Measures and Requirements

For the ActEV 2019 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. For this challenge, real-time processing refers to processing at the same rate as the input video.

For the ActEV 2019 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 each task separately.

For ActEV 2019 Independent evaluation, NIST will run the challenge participants' system on one of the nodes of the NIST Independent Evaluation Infrastructure costing less than $10K (Hardware specification for ActEV 2019 are provided in Appendix C and more details for speed measures and requirements are in Appendix A).

6.5.3. Training Data and Knowledge Sources

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

6.5.4. System References

List pertinent references, if any.

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 ActEV2018 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 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 ActEV 2019 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 ActEV 2019 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.

For ActEV 2019 Independent evaluation, NIST will run the challenge participants system on one of the nodes of the NIST Independent Evaluation Infrastructure costing less than $10K (Hardware specification are provided in Appendix C). The systems may not be more than 20 times slower than realtime for 18 target activities.

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: SCHEMAS

JSON Schema for system output file

Please refer to the ActEV_Scorer software package (same for the ActEV 2019 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.

NIST Independent Evaluation Infrastructure Specification

Hardware specification:

- CPU - 16 cores
- Memory - 128GB

- GPU – 4 x NVIDIA® GeForce GTX 1080 Ti GPU - 11GB
- Root disk size - 40GB
- 250GB SSD cache
- Storage Volume- 1TB (variable)
- Supplied object store (read only) for source video

Independent Evaluation Infrastructure and Delivery of Software

The challenge participants will deliver their algorithms that are compatible with the CLI protocol to NIST. The purpose is to test for compatibility and to ensure that the test results that the challenge participants obtained on the "validation dataset" when running on their own server match what NIST is getting when they run it on the Independent Evaluation Infrastructure.

Appendix C: 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": "??????"}}}'
```

- Similarly download the MEVA dataset

Appendix D: Definitions of Activity and Required 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 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.

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.