TRECVID ActEV 2020 Evaluation Plan

Date: 2020-04-02

ActEV Team

NIST

Table of Contents

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 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 ActEV Sequestered Data Leaderboard (ActEV SDL) based on the Multiview Extended Video (MEVA) Test3 dataset [10] with 37 activities and with updated names. 2) The TRECVID 2020 ActEV self-reported leaderboard based on the VIRAT V1 and V2 datasets [9] with 35 activities and with updated names.

TRECVID ActEV 2020 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 2020 leaderboard evaluation.

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	person_opens_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 2020 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 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 2020 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 2020 evaluation, there will be two types of evaluation; a self-reported TRECVID ActEV 2020 leaderboard evaluation and an ActEV 2020 SDL independent evaluation for the selected participants.

2.3.1. TrecVID ActEV 2020 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/trecvid20).

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 website: https://actev.nist.gov/sdl

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 be used for online learning/adaptation during test data processing.

2.5. Required Evaluation Condition

For TRECVID ActEV 2020 Leaderboard evaluation, the conditions can be summarized as shown in Table below:

ActEV 2020 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 SDL 2020 Independent evaluation (https://actev.nist.gov/sdl), the conditions can be summarized as shown in Table below:

ActEV 2020 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	MEVA

3. Data Resources

This data used for TRECVID ActEV 2020 Leaderboard evaluation is the VIRAT V1 and V2 datasets and the ActEV SDL 2020 Independent evaluation is based on sequestered MEVA dataset.

The table below provides a list of activities for the TRECVID ActEV 2020 evaluation. The 35 target activities are used in the ActEV 2020 leaderboard. The detailed definitions of the activities and its associated objects are described in the Annotation_Guide_lines doc: https://gitlab.kitware.com/viratdata/viratannotations/blob/master/DIVA-Annotation-Guidelines-V1.0.docx.pdf.

Table: List of activities for TRECVID ActEV 2020 with the new names and the original names.

VIRAT Activity Name (Original)	**VIRAT Activity Name 2020**
Closing	person_closes_facility_or_vehicle_door
Closing_Trunk	person_closes_trunk
DropOff_Person_Vehicle	vehicle_drops_off_person
Entering	person_enters_facility_or_vehicle
Exiting	person_exits_facility_or_vehicle
Interacts	person_interacts_object
Loading	person_loads_vehicle
Open_Trunk	person_opens_trunk
Opening	person_opens_facility_or_vehicle_door
Person_Person_Interaction	person_person_interaction
PickUp	person_pickups_object
PickUp_Person_Vehicle	vehicle_picks_up_person
Pull	person_pulls_object

Push	person_pushs_object
Riding	person_rides_bicycle
SetDown	person_sets down_object
Talking	person_talks_to_person
Transport_HeavyCarry	person_carries_heavy_object
Unloading	person_unloads_vehicle
activity_carrying	person_carries_object
activity_crouching	person_crouches
activity_gesturing	person_gestures
activity_running	person_runs
activity_sitting	person_sits
activity_standing	person_stands
activity_walking	person_walks
specialized_talking_phone	person_talks_on_phone
specialized_texting_phone	person_texts_on_phone
specialized_using_tool	person_uses_tool
vehicle_moving	vehicle_moves
vehicle_starting	vehicle_starts
vehicle_stopping	vehicle_stops
vehicle_turning_left	vehicle_turns_left
vehicle_turning_right	vehicle_turns_right
vehicle_u_turn	vehicle_makes_u_turn

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,
     "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)

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

- 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)

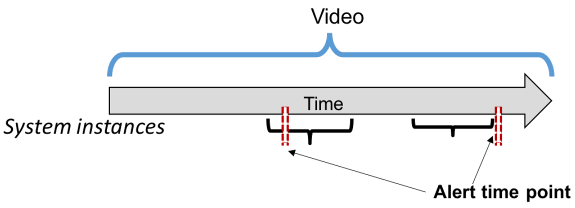

Figure 1. Example of system reported alert time

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 SDL 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:

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

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

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 the system 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.

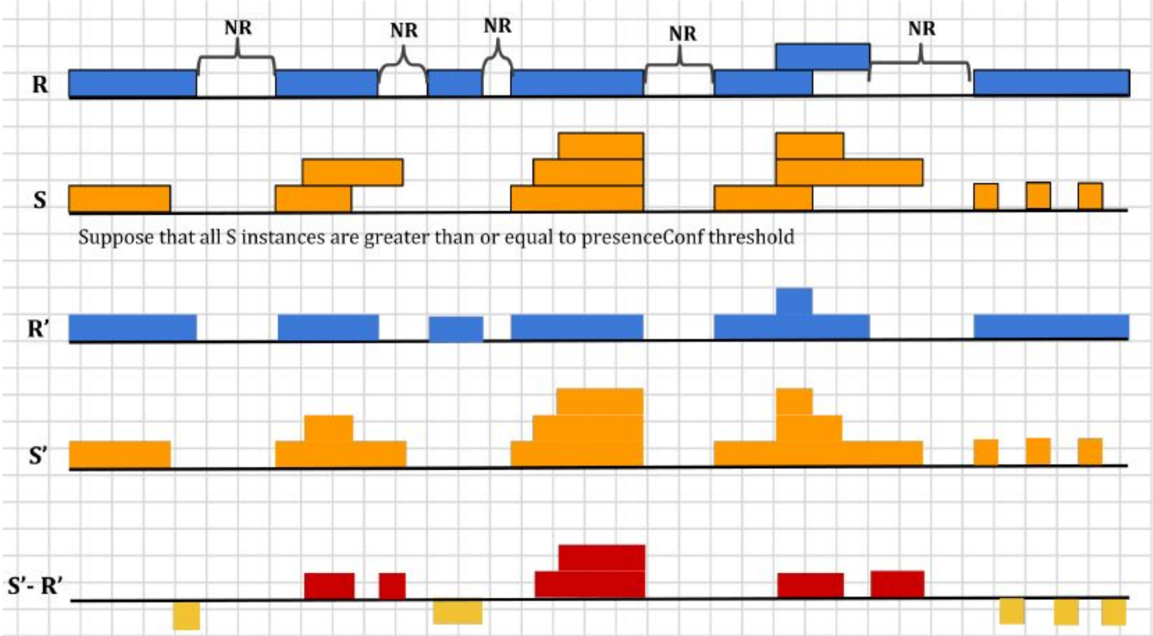
NR
NR
NR
NR
NR
R
S
Suppose that all S instances are greater than or equal to presenceConf threshold
R'
S'
S'- R'

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 a 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

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

$$\text{when } I_{R_i} >= 1 \text{ sec, } \varnothing \text{ } if\ Intersection(I_{R_i}, I_{S_j}) < 1 \text{ sec,}$$

$$\text{when } I_{R_i} < 1 \text{ sec, } \varnothing \text{ } if\ Intersection(I_{R_i}, I_{S_j}) < 50\%\ of\ I_{R_i}\ time$$

$$1 + AP_{con}(I_{S_j}), \quad otherwise\}$$

where,

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

[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.

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.

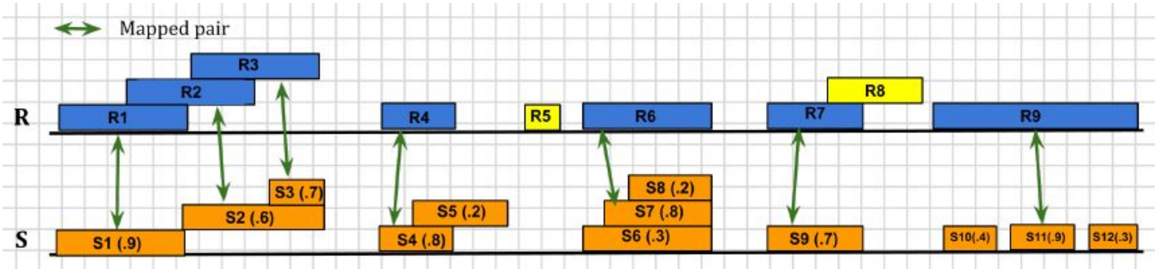

Figure 2: Pictorial depiction of activity instance alignment and P_{miss} calculation
(In *S*, the first number indicates instance id and the second indicates *presenceConf* score. For example, S1 (.9) represents the instance S1 with corresponding confidence score 0.9. Green arrows indicate aligned instances 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[3] is:

```
% ActEV_Scorer.py Actev_SDL_V2 -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 Actev_SDL_V2 -s system-output.json -a activity-index.json -f
    file-index.json -F -v -V
```

7. System Information

7.1. System Description

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

7.2. 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.

7.2.1. Speed Measures and Requirements

For the ActEV 2020 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 2020 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.

7.2.3. Training Data and Knowledge Sources

[3] (Dec 16th, 2019) The ActEV Scorer was updated with a new scoring protocol Actev_SDL_V2 that changes the rule for backing off to 50% if the ref instance duration is less than 1 sec. Please do a git pull to get the lastest code

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

7.2.4. 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 string 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 2020 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 2020 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: SCHEMAS

JSON Schema for system output file

Please refer to the ActEV_Scorer software package (same for the ActEV 2020 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 C: Data Download

VIRAT Video Dataset

The VIRAT Video Dataset is designed to be realistic, natural and challenging for video surveillance domains in terms of its resolution, background clutter, diversity in scenes, and human activity/event categories than existing action recognition datasets. It has become a benchmark dataset for the computer vision community. Please download the videos from viratdata.org. The evaluation will be based on 35 activities from the activities, see activities tab on the trecvid20 website: https://actev.nist.gov/trecvid20#tab_activities

ActEV Data GIT repo access : See actev-data-repo: https://gitlab.kitware.com/actev/actev-data-repo

This GIT Repo is the data distribution mechanism for the ActEV evaluation. The repo presently consists of a collection of corpora and partition definition files to be used for evaluations. The new training and validation annotations for the 35 activities with the names will be available soon. The repo contains textual data but not the large-sized corpora (videos, etc.). Please download the videos from viratdata.org

Create a login account by registering (https://actev.nist.gov/trecvid20) for the TRECVID ActEV 2020. During account registration, you will:

- acknowledge that you have read and accepted the VIRAT data license
- agree to the rules of the TRECVID 2020 ActEV Evaluation Plan

You will then be able to make submissions. If there is any issue please email us at actev.nist@nist.gov

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.