

ActEV 2020 Sequestered Data Leaderboard (SDL) Evaluation Plan

(<https://actev.nist.gov/sdl>)

Updates:

- MEVA Test3 Dataset
- The new activity names
- Data now supports multiview

Date: Feb 28, 2020

ActEV Team, NIST

Contact: actev-nist@nist.gov

TABLE OF CONTENTS

Updates	3
1. Overview	3
2. Evaluation Task and Conditions	4
2.1. Task Definition	4
2.2. Conditions	5
2.3. Evaluation Type	5
2.4. Protocol and Rules	5
2.5. Required Evaluation Condition	5
3. Multiview data support	6
4. Data Resources	6
4.1. The Training/development resources (as of March 02, 2020)	6
4.2. Sequestered Evaluation Data (as of March 02, 2020)	7
4.3. Activity Definitions and annotations	7
5. System Input	8
6. System Output	9
6.1. System Output File for Activity Detection Tasks	9
6.2. Validation of Activity Detection System Output	10
7. Activity Detection Metrics	10
7.1. Computation of Time-Based False Alarm	12
7.2. Runtime Speed Calculations	13
7.3. Time-Limited Scoring	13
7.3. Alignment used in Computation of Probability of Missed Detection	15
7.4. ACTEV_Scoring Command Line	18
APPENDIX	18
Appendix A: NIST Independent Evaluation Infrastructure Specification	18
Appendix B: ActEV Command Line Interface for Software Delivery	18
Appendix C: Data Download	19
References	19
Disclaimer	20

Updates

- March 01, 2020: The ActEV SDL open with the expanded MEVA Test3 dataset
- March 01, 2020: We have updated the activity names for the SDL
- March 01, 2020: The data now supports multiview
- Feb 22, 2020: The ActEV Scorer has been updated (pp. 14-15)
- Jan 11, 2020: The updated ActEV scorer deployed

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 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 38 activities and with updated names.

The ActEV SDL 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. Participants are 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. See Appendix B for pointers to the ActEV CLI.

Challenge participants will develop activity detection and temporal localization algorithms for 37 activities that are to be found in extended videos and video streams.

The ActEV SDL evaluation is based on the Multiview Extended Video with Activities (MEVA) dataset. For ActEV participants, the MEVA dataset is available for download without a fee at mevadata.org (mevadata.org). Please click on the data tab of the ActEV SDL website (actev.nist.gov/sdl) for more information on the data.

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”. Detailed activity definitions are in the ActEV Annotation Definitions for MEVA Data document [7]. 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. Evaluation Task and Conditions

2.1. TASK DEFINITION

In the ActEV SDL 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 SDL 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

The participants will provide their runnable system to NIST using the Evaluation Container Submission Instructions [see details in Appendix B] for independent (sequestered) evaluation. The system will be run and evaluated on the MEVA sequestered data using NIST hardware--see the details in Appendix A for the hardware infrastructure.

2.4. PROTOCOL AND RULES

During the ActEV SDL evaluation, each participant may submit a maximum of one CLI system per week.

System runtime must be less than or equal to $1 \times$ the data length [see section 7.2].

2.5. REQUIRED EVALUATION CONDITION

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

ActEV SDL Independent	Required
Task	Activity Detection
Target Application	Forensic Systems
Evaluation Type	Sequestered Evaluation

Submission	See the details in Appendix A for Submission Instructions
Dataset	MEVA

3. Multiview data support

With the MEVA Test3 dataset there will be support for multiview data. More information will be coming soon.

4. Data Resources

The ActEV SDL evaluation is based on the Multiview Extended Video with Activities (MEVA) dataset (mevadata.org) collected at the Muscatatuck Urban Training Center with a team of over 100 actors performing in various scenarios. The data was built by the Intelligence Advanced Research Projects Activity (IARPA) Deep Intermodal Video Analytics (DIVA) program to support activity detection in multi-camera environments for both DIVA performers and the broader research community. There is a MEVA data users Google group to facilitate communication and collaboration for those interested in working with the data (meva-data-users group).

The MEVA dataset has two parts: the public training and development data and sequestered evaluation data used only by NIST to test systems. The data is accompanied by activity annotations.

4.1. THE TRAINING/DEVELOPMENT RESOURCES (AS OF MARCH 02, 2020)

The Multiview Extended Video with Activities (MEVA) dataset website (mevadata.org) is to share the public MEVA video dataset and annotations. The size of the public MEVA dataset is 333 hours of ground-camera and UAV video. The size of the provided annotated video dataset is 28 hours. ActEV participants are encouraged to annotate the MEVA KF1 dataset for the 37 activities as described at mevadata.org.

The MEVA data GIT repo is the data distribution mechanism for MEVA Related annotations and documentation. The repo presently consists of schemas for the activity annotations (<https://gitlab.kitware.com/meva/meva-data-repo>).

The ActEV data 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 the evaluations (<https://gitlab.kitware.com/actev/actev-data-repo>).

4.2. SEQUESTERED EVALUATION DATA (AS OF MARCH 02, 2020)

As of March 2020, NIST is using a 140-hour collection of annotated MEVA data for sequestered data evaluations. The data set consists of both EO and IR cameras, public cameras (examples of which are in the public data set). The leaderboard presents results on the full 140-hour collection reporting separately by EO and IR data. Developers receive additional scores by activity for the EO_subset1 and the IR_subset1. Both subsets consists of data from public cameras.

4.3. ACTIVITY DEFINITIONS AND ANNOTATIONS

The activity names were updated February 2020 for the SDL evaluation and the new names will be used for the duration of the ActEV program. The table below shows the "ActEV 2020 SDL Activity Name" and "ActEV 2019 SDL Activity Name" of the 37 activities to be detected for the ActEV SDL evaluation. The detailed activity definitions are in the ActEV Annotation Definitions for MEVA Data document (<https://gitlab.kitware.com/meva/meva-data-repo/blob/master/documents/MEVA-Annotation-Definitions.pdf>). The activity name mapping from 2019 to 2020 file is (<https://gitlab.kitware.com/meva/meva-data-repo/-/blob/master/documents/activity-name-mapping.csv>).

ActEV 2020 SDL Activity Name	ActEV 2019 SDL Activity Name (DEPRECATED)
person_abandons_package	abandon_package
person_closes_facility_door	person_closes_facility_door
person_closes_trunk	Closing_Trunk
person_closes_vehicle_door	person_closes_vehicle_door
person_embraces_person	person_person_embrace
person_enters_scene_through_structure	person_enters_through_structure
person_enters_vehicle	person_enters_vehicle
person_exits_scene_through_structure	person_exits_through_structure
person_exits_vehicle	person_exits_vehicle
hand_interacts_with_person	hand_interaction
person_carries_heavy_object	Transport_HeavyCarry
person_interacts_with_laptop	person_laptop_interaction
person_loads_vehicle	person_loads_vehicle

person_transfers_object	object_transfer
person_opens_facility_door	person_opens_facility_door
person_opens_trunk	Open_Trunk
person_opens_vehicle_door	person_opens_vehicle_door
person_talks_to_person	Talking
person_picks_up_object	person_picks_up_object
person_purchases	person_purchasing
person_reads_document	person_reading_document
person_rides_bicycle	Riding
person_puts_down_object	person_sets_down_object
person_sits_down	person_sitting_down
person_stands_up	person_standing_up
person_talks_on_phone	specialized_talking_phone
person_texts_on_phone	specialized_texting_phone
person_steals_object	theft
person_unloads_vehicle	Unloading
vehicle_drops_off_person	vehicle_drops_off_person
vehicle_picks_up_person	vehicle_picks_up_person
vehicle_reverses	vehicle_reversing
vehicle_starts	vehicle_starting
vehicle_stops	vehicle_stopping
vehicle_turns_left	vehicle_turning_left
vehicle_turns_right	vehicle_turning_right
vehicle_makes_u_turn	vehicle_u_turn

5. System Input

The subset of video files to be process for an evaluation will be specified by a set of two files: an ActEV Evaluation “file index” JSON file that specifies the video files to be processed and metadata about the video including metadata supporting multiview processing systems, and an ActEV Evaluation “activity index” JSON file that specifies the activity names the tested system is expected to detect. Both formats are described in the ActEV Evaluation JSON Formats Document [8].

6. System Output

In this section, the system output format is defined. The ActEV Scorer software package¹ 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.

6.1. 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": [
    "2018-03-07.16-50-00.16-55-00.hospital.G479.avi"
  ],
  "activities": [
    {
      "activity": "Talking",
      "activityID": 1,
      "presenceConf": 0.89,
      "localization": {
        "2018-03-07.16-50-00.16-55-00.hospital.G479.avi": {
          "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:
 - o activity: (e.g. "Talking")

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

- o 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)
- o 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.
- o 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>
 - <framenumbers>: 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)

6.2. VALIDATION OF ACTIVITY DETECTION SYSTEM OUTPUT

To use the ActEV_Scorer ² to validate system output “SYSTEM.json”, execute the following command:

```
% ActEV_Scorer.py Actev_SDL_V2 -V -s SYSTEM.json -a activity-index.json -f
file-index.json
```

7. 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 $nAUDC_a$.

² (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 latest code

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:

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

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

7.1. 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.

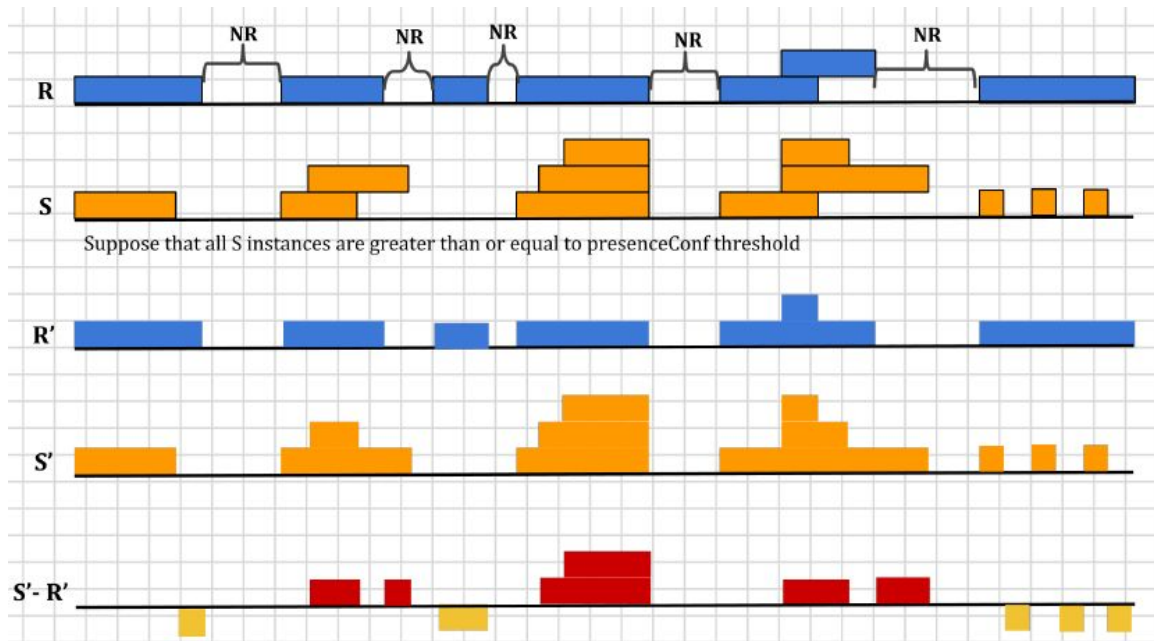


Figure 1: Pictorial depiction of T_{fa} calculation

(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).

7.2. RUNTIME SPEED CALCULATIONS

ActEV SDL systems are expected to process video in less than real time compared to the video duration of the test material. NIST will calculate runtime speed by capturing execution durations based on a subset of ActEV Command Line Interface (ActEV CLI) calls with the intent to exclude processing times before the system receives access to the test video and excludes time taken to shut down the instance.

The SDL Execution system distributes the execution across multiple nodes by dividing the data set into independent sub-parts. Beginning in February 2020, the approximate video duration per part is 2 hours. For each sub part, NIST will collect processing times from the following ActEV CLI calls:

- *actev-design-chunks*
- *actev-experiment-init*
- *actev--pre-process-chunk*
- *actev-process-chunk*
- *actev-post-process-chunk*
- *actev-merge-chunk*
- *actev-experiment-cleanup*

The Real Time Factor (RTFactor) then computed by aggregated over sub parts (Ns):

$$RTFactor = SyTime / Video_duration$$

where

$$SyTime = \sum_1^{Ns} (\text{durations of CLI calls above})$$

RTFactor will be computed and reported separately for EO and IR videos.

Note: RTFactors reported through February 2020 were aggregated over the entire collection. This will change with updates during March 2020.

7.3. TIME-LIMITED SCORING

If an SDL system takes longer than realtime process videos NIST will rescore the system as if to simulate what the score would be if system execution as stopped in the middle of the sub part when the runtime exceeded realtime. This means the system will incur miss detections for the stopped sub-part and all subsequent sub parts.

The leaderboard will report two time-limited metrics are added to show how systems would perform in realtime.

1. Time limited partial AUDC
2. Time limited mean-p_miss@0.04 TFA

Note: The leaderboard scores as of February 2020 do not fully report the runtime speeds separately for EO and IR because of various issues. The reported runtime speeds are aggregated over EO and IR and the time-limit performances scores are approximated for EO and IR. This will be changed during revisions in March 2020.

7.3. 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 instances³. 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

³ 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.

$$K(I_{R_i}, I_{S_j}) = \{\emptyset \text{ if } Activity(I_{S_j}) \neq Activity(I_{R_i})$$

$$\text{when } I_{R_i} \geq 1 \text{ sec, } \emptyset \text{ if } Intersection(I_{R_i}, I_{S_j}) < 1 \text{ sec,}$$

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

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

where,

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

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; \emptyset 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.

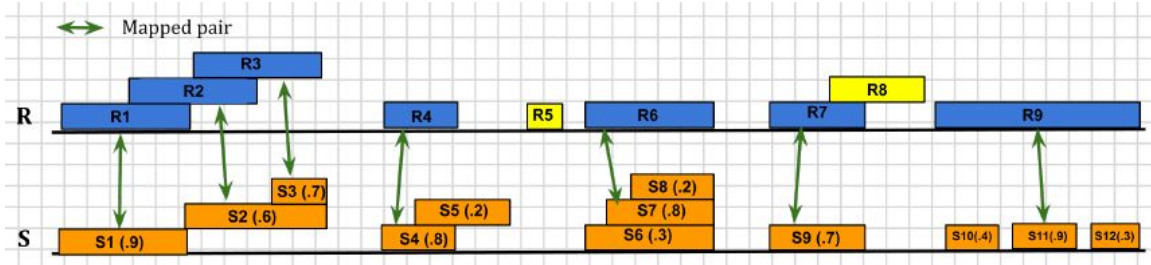


Figure 2: Pictorial depiction of activity instance alignment and P_{miss} calculation (In S , the first number indicates instance id and the second indicates *presenceConf* score. For example, S1 (.9) represents the instance S1 with corresponding confidence score 0.9. Green arrows indicate aligned instances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

7.4. ACTEV_SCORING COMMAND LINE

The command to score a system using the ActEV_Scorer⁴ 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
```

APPENDIX

APPENDIX A: NIST INDEPENDENT EVALUATION INFRASTRUCTURE SPECIFICATION

Hardware specification:

- Chassis: Asus ESC4000 G4S
- CPU: 2x Intel(R) Xeon(R) Silver 4214 CPU @ 2.20GHz
- Motherboard: Asus Intel® C621 PCH chipset
- HDD/SSD: 2x 1.92TB Intel SSD DC S4500
- RAM: 12x 16GB DDR4-2400 ECC RDIMM
- GPU: PNY RTX2080Ti blower style
- OS: Ubuntu 18.04
- Storage Volume- 1TB (variable)
- Supplied object store (read only) for source video

APPENDIX B: ACTEV COMMAND LINE INTERFACE FOR SOFTWARE DELIVERY

The challenge participants will deliver their algorithms that are compatible with the ActEV Command Line Interface (ActEV CLI) protocol to NIST. The CLI

⁴ (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 latest code

documentation prescribes the steps to install the software/algorithm from a web-downloadable URL and run the algorithm on a video dataset. The steps include downloading software and models, installing 3rd party packages, testing the software on a validation data set, processing video through the system, and delivering system output. For more information on the ActEV Command Line Interface (ActEV CLI) for the ActEV SDL evaluation, please visit the “Algorithm Submission” tab on the ActEV SDL website (<https://actev.nist.gov/sdl>).

APPENDIX C: DATA DOWNLOAD

You can download the MEVA video and annotations dataset for free from the mevadata.org website (<http://mevadata.org/>)

To download all the other data, visit the data tab on the ActEV SDL evaluation website (<https://actev.nist.gov/sdl>).

Then 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 update a previously downloaded repo with 'git pull'. Note: this is the same repo as used from VIRAT.
 - Clone: `git clone https://gitlab.kitware.com/actev/actev-data-repo.git`
 - Update: `cd "Your_Directory_For_actev-data-repo"; git pull`

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] ActEV Annotation Definitions for MEVA Data document:
<https://gitlab.kitware.com/meva/meva-data-repo/blob/master/documents/MEVA-Annotation-Definitions.pdf>
- [8] ActEV Evaluation JSON Formats document:
https://gitlab.kitware.com/meva/meva-data-repo/-/blob/master/documents/nist-json-for-actev/ActEV_Evaluation_JSON.pdf
- [9] VIRAT Video Dataset: <http://www.viratdata.org/>
- [10] Multiview Extended Video (MEVA) dataset: <http://mevadata.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.