

ActEV 2021 Sequestered Data  
Leaderboard (SDL)  
Evaluation Plan

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

**Date: April 16, 2021**  
**ActEV Team, NIST**  
**Contact: [actev-nist@nist.gov](mailto:actev-nist@nist.gov)**



## Updates

- September 17th, 2020:
  - Added new Unknown Facilities (UF) leaderboards for both Known Activities (KA) and Surprise Activities (SA)
  - Updated CLI implementation and documentation to support training surprise activities and unknown facilities data
  - Modified system input and output JSON formats to accommodate Surprise Activity evaluation
- September 21st, 2020
  - Opened ActEV2021 SDL UF with KA submissions
- September 25th, 2020
  - Opened ActEV2021 SDL KF with KA submissions
- November 2nd, 2020
  - Opened submissions of ActEV'21 SDL UF with SA for NIST-internal debugging
- November 20th, 2020
  - Opened submissions of ActEV'21 SDL UF with SA for automation debugging
- December 4th, 2020
  - Opened ActEV'21 SDL UF with SA in fully automated pipelines
- April 16, 2021
  - Added more detail to Section 6.1

## TABLE OF CONTENTS

1. Overview	4
2. Evaluation Task and Conditions	5
2.1. Task Definition	5
2.2. Conditions	5
2.3. Evaluation Type	5
2.4. Protocol and Rules	6
2.5. Required Evaluation Condition	6
3. Multiview data support	6
4. Data Resources	7
4.1. The Training/development resources	7
4.2. Sequestered Evaluation Data	8
4.3. Activity Definitions and annotations	8
5. System Input	9
6. System Output	9
6.1. System Output File for Activity Detection Tasks	10
6.2. Validation of Activity Detection System Output	12
7. Performance Metrics for Activity Detection	12
7.1. Computation of Time-Based False Alarm	13
7.2. Runtime Speed Calculations	14
7.2. Surprise Activity Training Speed Calculations	15
7.3. Time-Limited Scoring	15
7.4. Alignment used in Computation of Probability of Missed Detection	16
7.5. 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

## 1. Overview

The Activities in Extended Video (ActEV) series of evaluations are designed to accelerate development of robust, multi-camera, automatic activity detection systems in known and unknown facilities for forensic and real-time alerting applications. 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.

ActEV began with the Summer 2018 self-reported and blind leaderboard evaluations and has currently progressed to the running of The ActEV 2021 Sequestered Data Leaderboard (SDL) evaluations (ActEV'21 SDL). The SDL evaluations include two separate evaluations: 1) Known Facility leaderboard based on the MEVA (mevadata.org) dataset and 2) Unknown Facility (UF) leaderboard for 37 known activities. The UF leaderboard also includes Surprise Activities that are evaluated on a separate sub-leaderboard. Submissions that address a sub-leaderboard with either known activities or surprise activities may be made and no-score will be displayed for the sub-leaderboard that is not addressed by the system.

For the **Known Facility** (KF) test, developers are provided metadata (when it exists) including the site map with approximate camera locations and sample FOVs, camera models and a 3D site model. Developers are allowed to use such metadata information during their system development. KF systems will be tested on both Electro-Optical (EO) and Infrared (IR) camera modalities. For the **Unknown Facility** (UF) test, systems are provided a subset of the KF metadata information and the metadata is only delivered during test time (during system execution). UF systems will be tested only on EO video modalities.

For the **Known Activities** (KA) tests, developers are provided a list of activities in advance for use during system development (e.g., training) for the system to automatically detect and localize all instances of the activities. On the other hand, for the **Surprise Activities** (SA) tests, the pre-built system is provided a set of activities with training materials (text description and at least one exemplar video chip) during system test time (in execution) to automatically develop detection and localization models. Then the system must automatically detect and localize all instances of the activities. To facilitate activity training at test time, systems will be provided a maximum of 10 hours to train for SAs while executing on the NIST hardware (See Appendix A: 2080 hardware for the UF leaderboards.)

For the ActEV'21 SDL evaluation, participants are required to submit their runnable activity detection software using the ActEV Command Line Interface (CLI) as described in Appendix B. NIST will then evaluate system performance on sequestered data using NIST hardware (see Appendix A) and results will be posted to a public leaderboard.

Submissions to the ActEV'21 SDL will be ranked independently within different leaderboards. There are two leaderboards (KF and UF) with sub-leaderboards as follows:

- UF: Unknown Facilities Leaderboards
  - UF Known Activities on the EO modality
  - UF Surprise Activities on the EO modality
- KF: Known Facilities Leaderboards
  - KF Known Activities on the EO modality
  - KF Known Activities on the IR modality

Participation in all the leaderboard is optional, however, the Unknown Facilities testing is the focus of the evaluation epoch.

The remainder of this document is organized as follows. Section 2 defines the evaluation tasks and conditions and Section 3 describes the multiview data support. Data resources, system inputs and outputs are given Section 4 through 6, respectively. Section 7 defines the performance metrics for activity detection. The detailed descriptions for NIST hardware, CLI, and data download are found in appendices.

## 2. Evaluation Task and Conditions

### 2.1. TASK DEFINITION

In the ActEV'21 SDL evaluation, there is one Activity Detection (AD) task for detecting and localizing activities. 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. The minimum temporal overlap with a single reference instance in the evaluation is 1 second. If the reference instance duration is less than 1 second, 50% of the reference duration is required as the minimum temporal overlap.

The AD task applies to both known and unknown facilities as well as both known and surprise activities.

### 2.2. CONDITIONS

The ActEV'21 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 ActEV'21 SDL tests are a sequestered data leaderboard evaluation. 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 KF and UF sequestered data using NIST hardware (see details in Appendix A.)

## 2.4. PROTOCOL AND RULES

During the ActEV'21 SDL evaluation, each participant may submit a maximum of one CLI system per week.

System runtime must be less than or equal to 1x the data length and the Surprise Activity training step must take less than 10 hours on the NIST Evaluation Server Hardware (see Section 7.2 for the details).

## 2.5. REQUIRED EVALUATION CONDITION

As described above, the ActEV'21 SDL evaluations include two leaderboards for two separate evaluations: 1) the Known Facility (KF) and 2) the Unknown Facility (UF) leaderboards. For each submission on the SDL, the participant will select which type of facility (either KF or UF) to test the system on. If a participant wishes to test a system on both KF and UF, the system must be submitted twice, once for each facility.

Submissions made to the KF Leaderboard will be tested on both EO and IR modality data. Submissions may produce no activity detections (and will garner a  $P_{miss} = 1$ ) for unsupported modalities and must be sufficiently robust to not crash. Participants should include verbiage in the "System Description" submission field documenting unsupported modalities.

Likewise, submissions made to the UF Leaderboard will be tested on both Known and Surprise Activities. Submissions may produce no activity detections (and will garner a  $P_{miss} = 1$ ) for unsupported activity and must be sufficiently robust to not crash. Participants should include verbiage in the "System Description" submission field documenting unsupported activity types.

## 3. Multiview data support

The test sets for both KF and UF leaderboards support testing systems capable of leveraging overlapping camera views to improve detection performance. At test time, these "Multiview Capable" systems will be given all recordings with overlapping fields of view so that the system can leverage the overlapping fields of view.

Testing "Multiview Capable" systems greatly increases the duration of the test collection by adding additional views beyond that used for the actual evaluation. If a system is not multiview capable, leave the "Multiview Capable" check box un-checked so that SDL computation resources are not wasted processing non-evaluation data.

## 4. Data Resources

The ActEV'21 SDL evaluation is based on the Known Facilities (KF) and Unknown Facilities (UF) datasets. The KF data was collected at the Muscatatuck Urban Training Center (MUTC) with a team of over 100 actors performing in various scenarios. The KF dataset has two parts: 1) the public training and development data and 2) sequestered evaluation data used by NIST to test systems. The UF data was collected at different places and includes both known and surprise activities.

The KF and UF data were collected and annotated for the Intelligence Advanced Research Projects Activity (IARPA) Deep Intermodal Video Analytics (DIVA) program. A primary goal of the DIVA program is 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) and the MEVA Public data can be found on the website (mevadata.org.)

### 4.1. THE TRAINING/DEVELOPMENT RESOURCES

The KF training and development data has been publicly released as the Multiview Extended Video with Activities (MEVA) dataset. Details for downloading the dataset and a link to a repository of associated activity annotations are available at the website [mevadata.org](http://mevadata.org). As of December 2019, a total of 328 hours of ground-camera data and 4.2 hours of Unmanned Aerial Vehicle (UAV) video have been released. In addition, 28 hours of the ground camera video have been annotated by the same team that has annotated the ActEV test set. Additional annotations have been performed by the public and are also available in the annotation repository. ActEV participants are encouraged to annotate the MEVA KF dataset for the 37 activities as described at [mevadata.org](http://mevadata.org) and post them to the annotation repository.

The MEVA data GIT repo (<https://gitlab.kitware.com/meva/meva-data-repo>) is the data distribution mechanism for MEVA-related annotations and documentation. The repo is the authoritative source for MEVA video and annotations. The repo presently consists of schemas for the activity annotations, annotations of the 37 activities of interest, and metadata. The repo also contains third party annotations donated by the community.

The ActEV data GIT repo (<https://gitlab.kitware.com/actev/actev-data-repo>), is the data distribution mechanism for the ActEV evaluation-related materials. The ActEV evaluation makes use of multiple data sets. This repo is a nexus point between the ActEV evaluation and the utilized data sets. The repo consists of partition definitions (e.g., train or validation) to be used for the evaluations.

## 4.2. SEQUESTERED EVALUATION DATA

The KF test set is a 140-hour collection of videos which consists of both EO and IR camera modalities, public cameras (video from the cameras and associated metadata are in the public training set) and non-public cameras (video is not provided on metadata.org and camera parameters are only provided to the systems at test time). The KF leaderboard presents results on the full 140-hour collection reporting separately for EO and IR data. Developers receive additional scores by activity for the EO\_Set1 and the IR\_Set1. The EO-Set1 and IR-Set1 are subsets of the entire test sets. For example, EO-Set1 is a random 50% of the EO data from public cameras and likewise for IR-Set1.

The UF test set has a large collection of videos exclusively in the EO spectrum (as of September 9, 2020). The UF leaderboard presents results separately for known activity and surprise activity types. The detailed information regarding activity types is discussed in the following section.

## 4.3. ACTIVITY DEFINITIONS AND ANNOTATIONS

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 known activity definitions and annotations are found in the “DIVA ActEV Annotation Definitions for MEVA Data” document [7]. Each activity is formally defined by four text elements:

<b>Element</b>	<b>Meaning</b>	<b>Example Definition</b>
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

For the Surprise Activities, systems will also be given exemplar chip-videos (at least one and potentially unbounded) which can be found in the GIT repo (<https://gitlab.kitware.com/actev/actev-data-repo>). The exemplars contain a single instance of the activity along with a frame-varying and single bounding box annotation that surrounds the entities (people and/or objects) that are involved in the target activity. The chip videos will be extracted from MEVA-style video (to be in the same domain as the test material) and be padded both temporally and spatially to provide a full view of the instance.

The table below shows the names of the 37 Known Activities for ActEV’21 SDL while the Surprise Activities will be kept sequestered.



### ActEV'21 SDL Known Activity Names

hand_interacts_with_person	person_sits_down
person_carries_heavy_object	person_stands_up
person_closes_facility_door	person_talks_on_phone
person_closes_trunk	person_talks_to_person
person_closes_vehicle_door	person_texts_on_phone
person_embraces_person	person_transfers_object
person_enters_scene_through_structure	person_unloads_vehicle
person_enters_vehicle	vehicle_drops_off_person
person_exits_scene_through_structure	vehicle_makes_u_turn
person_exits_vehicle	vehicle_picks_up_person
person_loads_vehicle	vehicle_reverses
person_opens_facility_door	vehicle_starts
person_opens_trunk	vehicle_stops
person_opens_vehicle_door	vehicle_turns_left
person_picks_up_object	vehicle_turns_right
person_puts_down_object	person_abandons_package
person_reads_document	person_interacts_with_laptop
person_rides_bicycle	person_purchases
	person_steals_object

## 5. System Input

The subset of video files to be processed for an evaluation will be specified by a set of two files: 1) an ActEV Evaluation “file index” JSON file that specifies the video files to be processed and metadata about the video (potentially including frame synchronizations) as described in the meadata.org documentation, and 2) an ActEV evaluation “activity index” JSON file that specifies the activity names the tested system is expected to detect. Both “file index” and “activity index” formats are described in the ActEV Evaluation JSON Formats Document (<https://gitlab.kitware.com/meva/meva-data-repo/-/tree/master/documents/nist-json-for-actev>).

## 6. System Output

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

---

<sup>1</sup>ActEV\_Scorer software package ([https://github.com/usnistgov/ActEV\\_Scorer](https://github.com/usnistgov/ActEV_Scorer))

## 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, a “processingReport” of file processing success and failures, and a collection of activity instance records with temporal localizations and spatial localizations of objects. DIVA performers sponsored by IARPA are required to provide spatial localizations as defined by the DIVA Program. Localizations provided by systems will not be utilized for computing leaderboard scores.

A notional system output file is included inline below, followed by a description of each field. See “ActEV Evaluation JSON Formats document” [8] for more specifics.

```
{
  "filesProcessed": [
    "2018-03-14.07-30-04.07-35-04.school.G336.avi",
    "2018-03-14.12-20-04.12-25-04.hospital.G436.avi", ...],
  "processingReport": {
    "fileStatuses": {
      "2018-03-14.07-30-04.07-35-04.school.G336.avi": {
        status: "success",
        message: "free text" },
      "2018-03-14.12-20-04.12-25-04.hospital.G436.avi": {
        status: "failed",
        message: "ffmpeg exited with non-zero error code" }, ...
    },
    "siteSpecific": { "content defined by the system" }
  },
  "activities": [
    {
      "activity": "Closing",
      "activityID": 1,
      "localization": {
        "2018-03-14.07-30-04.07-35-04.school.G336.avi": {
          "20": 1,
          "30": 0
        }
      }
    },
    "objects": [
      {
        "objectType": "person",
        "objectID": 1,
        "localization": {
          "2018-03-14.07-30-04.07-35-04.school.G336.avi": {
            "1": { "boundingBox": {"x":10, "y":30,"w":50,"h":20}},
            "20": { "boundingBox": {"x":10, "y":32,"w":50, "h":20}},,
            "30": { "boundingBox": {"x":10, "y":35,"w":53, "h":20}},
            "112": {}
          }
        }
      }
    ]
  }
}
```

- filesProcessed: An array enumerating the file names processed. Every file, even if the file was unreadable or contained no activities, must be present in the array. The “processingReport” dictionary below can be used to report anomalies.
- processingReport: A mandatory dictionary to report success or failures during the processing of the videos.
  - fileStatuses: A dictionary reporting success or failures while processing videos. The keys to the dictionary are the file names used in filesProcessed.
    - <filename>
      - status: A text string indicating success or failure. The values must be “success” or “fail”
      - message: An additional text string to report additional information. The content is not restricted.
  - siteSpecific: An optional dictionary for which the system can store additional information. The structure and content of this dictionary has no restrictions.
- activities: An array of annotated activity instances. Each instance is a dictionary with the following fields:
  - activity: The name (e.g. “Talking”) from the MEVA Annotation [7]
  - activityID: a unique, numeric identifier for the activity instance. The value must be unique within the list of activity detections for all video source files processed (i.e. within a single activities JSON file)
  - localization: The temporal localization of the activity instance encoded as a dictionary of frame state signals indexed by the video file id(s) for which the activity instance is observed. The each frame state signal has keys representing a frame number and the value being 1 (the activity instance is present) and 0 (otherwise) within the given video file. Multiple frame state signals can be used to represent an activity instance being present in multiple video views. In this case, frame numbers are relative with respect to the video file. The frame number begins at “1” which means frame 1 corresponds to the first frame.
  - objects: An array of objects annotated with respect to the activity instance. Each unique object is represented by the following dictionary:
    - objectType: A string identifying the objects type (e.g., person or object) as one of the track types defined in the MEVA Annotation Spec.
    - objectID: unique, numeric identifier for the object. The value must be unique within a single activities JSON file.
    - Localization: The temporal-spatial localization of the objectType referred to by the record encoded as a dictionary of frame state signals indexed by the video file id for which the object is observed. The frames for the object localization is not necessary to match the

frames for the activity localization. Each frame state signal (for a given video) has keys representing a frame number and the value is a dictionary describing the spatial localization of the object. The spatial dictionary is either empty, indicating no localization (used to terminate a track or indicate an object is not visible), or has 1 key 'boundingBox' which is itself a dictionary described as a pixel 'x', 'y', 'w', and 'h' for the the x-position, y-position, width and height respectively. The (0,0) (x,y) position is the top left pixel. Object localization for a frame state signal is interpreted to extend from the key value frame until the next frame present in the frame state signal, or until the end of the given video if no additional frames are present.

## 6.2. VALIDATION OF ACTIVITY DETECTION SYSTEM OUTPUT

To use the ActEV\_Scorer 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. Performance Metrics for Activity Detection

The technologies sought for the ActEV SDL leaderboard evaluation are expected to report activities that visibly occur in a single-camera video by specifying 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:

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

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

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

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

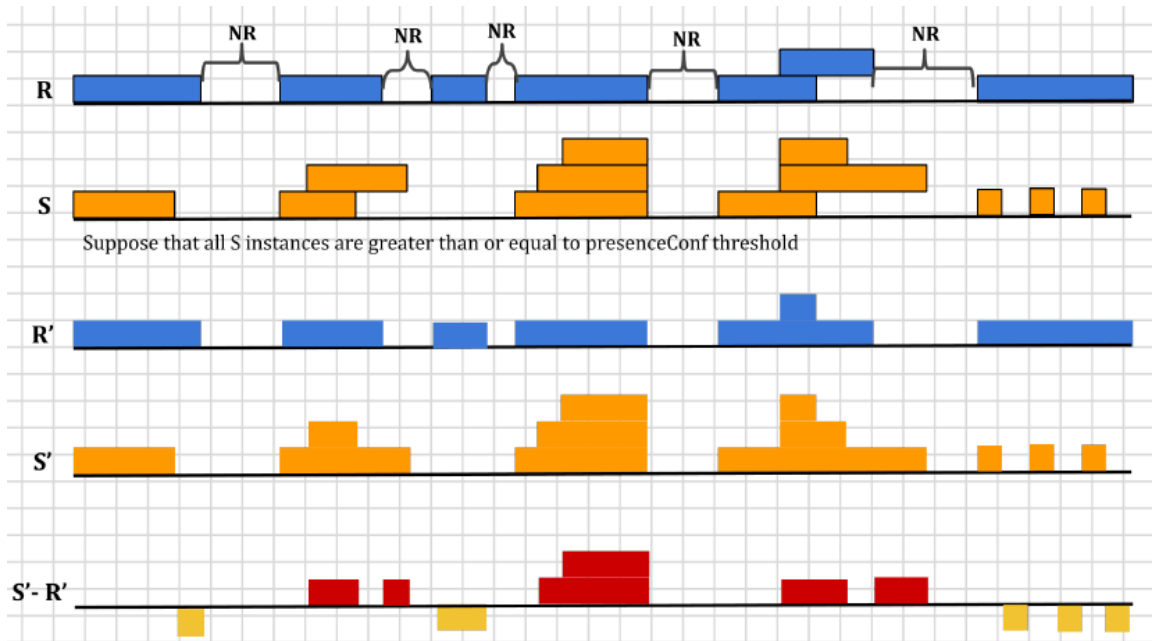
$N_{frames}$ : The duration (frame-based) of the video  
 $NR$ : Non-Reference duration. The duration of the video without the target activity occurring  
 $S'_i$ : the total count of system instances for frame  $i$   
 $R'_i$ : the total count of reference instances for frame  $i$   
 $T_{fa}$ : The time-based false alarm value (see Section 7.1 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 7.2 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 reference 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 activity time do not increase  $T_{fa}$  unless the detections 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 real time or quicker compared to the test set video duration. 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. For more information on APIs and CLIs for independent evaluation see the document

[https://gitlab.kitware.com/actev/diva\\_evaluation\\_cli/-/blob/master/doc/introduction.md](https://gitlab.kitware.com/actev/diva_evaluation_cli/-/blob/master/doc/introduction.md)

The SDL Execution system has the capability to perform system execution on the full data set (the SDL single partition flag) or by distributing execution across multiple nodes by dividing the data set into independent sub-parts (the SDL multi-partition flag). The approximate video duration per part is 2 hours. Regardless of the chosen execution method, 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 aggregating over sub parts ( $N_s$ ):

$$RTfactor = \frac{SysTime}{VideoDuration}$$

where

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

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

**Note:** *RTFactors* reported through February 2020 were aggregated over the entire collection.

## 7.2. SURPRISE ACTIVITY TRAINING SPEED CALCULATIONS

A system capable of detecting surprise activities will be provided *a maximum of 10 hours* to complete activity training on the test server. This time will be calculated by monitoring the duration of the ActEV CLI call ‘*actev-train-system*’. At NIST’s discretion, systems exceeding the 10-hour time limit will not proceed to the execution phase where test material is processed.

## 7.3. TIME-LIMITED SCORING

If an SDL system takes longer than real-time to process test videos, NIST will score the system as if system execution has stopped in the middle of the sub part where the runtime exceeded real time. This means the system will incur missed detections for the stopped sub-part and all subsequent sub parts.

The ActEV ’21 SDL Leaderboard will report only time-limited scores to show how systems would perform in real-time. The scores are:

1. Time limited partial AUDC ( $nAUDC_a$ ). This metric will be used for the public leaderboard ranking.
2. Time limited mean probability of missed detections at Time-based false alarm 0.02 ( $\mu P_{miss}@T_{fa} = 0.02$ ). This metric will be used for IARPA DIVA program ranking.

#### 7.4. 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<sup>2</sup>. 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.

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

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

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

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

<sup>2</sup> For instance, if there are two *person\_abandons\_package* 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;  $\emptyset$  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.

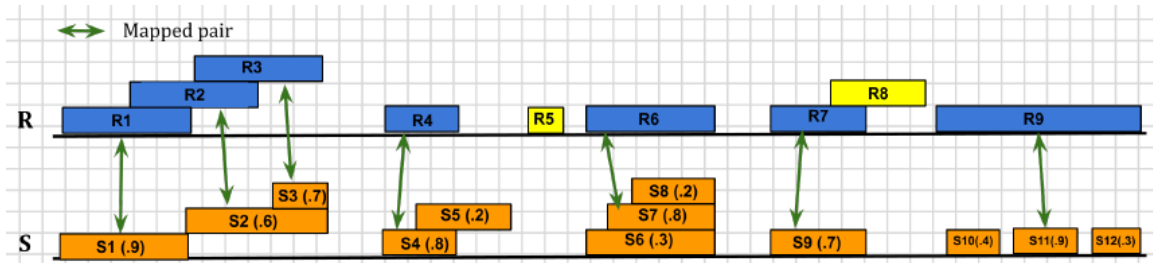


Figure 2: Pictorial depiction of activity instance alignment and  $P_{miss}$  calculation (In  $S$ , the first number indicates instance id and the second indicates *presenceConf* score. For example, S1 (.9) represents the instance S1 with corresponding confidence score 0.9. Green arrows indicate aligned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

## 7.5. ACTEV\_SCORING COMMAND LINE

The command to score a system using the ActEV\_Scorer 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
```

For more information on the scoring code, see the ActEV\_Scorer GIT Repo.  
([https://github.com/usnistgov/ActEV\\_Scorer](https://github.com/usnistgov/ActEV_Scorer))

## APPENDIX

### APPENDIX A: NIST INDEPENDENT EVALUATION INFRASTRUCTURE SPECIFICATION

Hardware specifications for ActEV'21 SDL evaluations are as follows.

1080 Hardware for the KF leaderboards:

- 32 VCPU 2.2Ghz (16 hyperthreaded)
- 128GB memory
- Root Disk: 40GB
- Ephemeral Disk: 256GB
- GPU: 4x - 1080Ti
- OS: Ubuntu 18.04

2080 Hardware for the UF leaderboards:

- 32 VCPU 2.2Ghz (16 hyperthreaded)
- 128GB memory
- Root Disk: 128GB
- Ephemeral Disk: 256GB
- GPU: 4x - 2080Ti
- OS: Ubuntu 18.04

### APPENDIX B: ACTEV COMMAND LINE INTERFACE FOR SOFTWARE DELIVERY

Leaderboard participants will deliver their algorithms that are compatible with the ActEV Command Line Interface (ActEV CLI) protocol to NIST. The CLI 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 public 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'.
  - Clone: `git clone https://gitlab.kitware.com/actev/actev-data-repo.git`
  - Update: `cd "Your_Directory_For_actev-data-repo"; git pull`
- Get an up-to-date copy of the [MEVA 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'.
  - Clone: `git clone https://gitlab.kitware.com/meva/meva-data-repo`
  - Update: `cd "Your_Directory_For_meva-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, 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* pp 1895-1898, 1997.
- [4] K. Bernardin and R. Stiefelhagen, “Evaluating multiple object tracking performance: The clear mot metrics,” *EURASIP Journal on Image and Video Processing*, 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](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.