ActEV 2021 Sequestered Data Leaderboard (SDL) Evaluation Plan

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

Date: April 28, 2021

Yooyoung Lee, Jonathan Fiscus, Andrew Delgado,
Lukas Diduch, Eliot Godard, Baptiste Chocot,
Jesse Zhang, Jim Golden, Afzal Godil, Diane Ridgeway

Contact: 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 20-27, 2021
    - Updated Section 5.1. System Output File for Activity Detection Tasks and JSON format example for localization
    - Added Appendix D: CLI Spatio-Temporal Localization Requirementt
    - Separate distinction of system output JSONs and reference annotation JSONs
    - Clarified several points about the phase 3 evaluation

TABLE OF CONTENTS

1. Overview

This document describes the system evaluations of the Activities in Extended Video (ActEV) sponsored as part of the Intelligence Advanced Research Projects Activity (IARPA) Deep Intermodal Video Analytics (DIVA) program. The ActEV evaluation plan covers task definition and condition, data resources, file formats for system inputs and outputs, performance metrics, scoring procedures, and protocol for submissions.

The ActEV evaluation series 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 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 ActEV 2021 Sequestered Data Leaderboard (SDL) evaluations. The ActEV21 SDL evaluations are summarized by facility, activity, and camera types in Table 1.

Table 1: SDL evaluations by Facility, Activity, and Camera Types

		Known Facility	Unknown Facility
Known Activities	EO (Electro-Optical)	√	√
	IR (Infrared)	√	
Surprise Activities	EO (Electro-Optical)		√

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.

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 clip) during system test time to automatically develop detection and localization models. The system must automatically detect and localize all instances of the activities.

For the **Known Facility** (KF) test, developers are provided metadata (when it exists) including the site map with approximate camera locations and sample Field of View (FOV), 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 known activity for 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 camera modalities for both known and surprise activities.

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 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 each test is optional, however, the Unknown Facilities testing is the focus of the evaluation.

The remainder of this document is organized as follows. Section 2 defines the evaluation tasks and conditions and Section 3 describes the data resources. Descriptions of system inputs and outputs are given Section 4 through 5, respectively. Section 6 defines the performance metrics for activity detection. The detailed descriptions for NIST hardware, CLI, and data download are found in appendices.

2. 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 an identified activity instance to be correct, the type of activity must be correct, and the temporal overlap must fall within a minimal requirement (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.

2.3. Evaluation Type

The ActEV'21 SDL tests are a sequestered data leaderboard evaluation. Participants will provide their system to NIST using the Evaluation Container Submission Instructions (see details in Appendix B) for 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

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.

During the evaluations, each participant may submit one CLI-compliant system per week. Additional submissions may be processed if additional resources are available.

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 6.2 for the details).

Participation in the evaluations open to all who find the task of interest. To fully participate a registered site must:

- become familiar with, and abide by, all evaluation protocol and submission;
- develop/enhance an algorithm that can process the required evaluation datasets;
- submit the necessary files to NIST for scoring (see the detailed submission steps at https://actev.nist.gov/sdl#tab_algo) ; and

Participants are free to publish results for their own system but must not publicly compare their results with other participants (ranking, score differences, etc.) without explicit written consent from the other participants.

While participants may report their own results, participants may not make advertising claims about their standing in the evaluation, regardless of rank, or winning the evaluation, or claim NIST endorsement of their system(s). The following language in the U.S. Code of Federal Regulations (15 C.F.R. § 200.113)14 shall be respected: NIST does not approve, recommend, or endorse any proprietary product or proprietary material. No reference shall be made to NIST, or to reports or results furnished by NIST in any advertising or sales promotion which would indicate or imply that NIST approves, recommends, or endorses any proprietary product or proprietary material, or which has as its purpose an intent to cause directly or indirectly the advertised product to be used or purchased because of NIST test reports or results.

At the conclusion of the evaluation, NIST may generate a report summarizing the system results for conditions of interest. Participants may publish or otherwise disseminate these charts, unaltered and with appropriate reference to their source.

The challenge participant can train their systems or tune parameters using any data complying with applicable laws and regulations.

2.5. MULTIVIEW 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.

2.6. EVALUATION CONDITION

As described above, the ActEV'21 SDL evaluations include two leaderboards along with sub-leaderboards: 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 Activities (KA) and Surprise Activities (SA). 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. Submissions that address a sub-leaderboard with either KA or SA may be made and no-score will be displayed for the sub-leaderboard that is not addressed by the system.

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.)

3. Data Resources

The ActEV'21 SDL evaluations are based on the KF and 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 IARPA 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.

3.1. THE TRAINING/DEVELOPMENT RESOURCES

The KF training and development data has been publicly released as the Multiview Extended Video with Activities (MEVA)[1] dataset. Details for downloading the dataset and a link to a repository of associated activity annotations are available at the website 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 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.

3.2. SEQUESTERED EVALUATION DATA

The KF test set is a 144-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 mevadata.org and camera parameters are only provided to the systems at test time). The KF leaderboard presents results on the full 144-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.

[1] 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.)

The UF test set has a large collection of videos exclusively in the EO spectrum. 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.

3.3. 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:

Table 2: An example of activity definition

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

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.

Table 3: 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

4. System Input

The system input includes a set of two files: 1) a"file index" JSON file that specifies the video files along with metadata (see the details in the documentation from mevadata.org., and 2) an "activity index" JSON file that specifies the activity names to be detected by a system. Both "file index" and "activity index" formats are described in the ActEV Evaluation JSON Formats Document [8].

5. System Output

This section describes the file format for the system output and its validation process.

5.1. Format of System Output File

The system output should be a JSON file that includes a list of videos processed by the system ("filesProcessed"), a report of the processing successes and failures ("processingReport"), and a collection of activity instance records with temporal localizations and spatial localizations of objects ("activities" and "objects")--see a JSON example in Table 4.

DIVA performers sponsored by IARPA are required to provide spatial localizations of objects as defined by the DIVA Program. Localization performance will NOT be displayed in the leaderboards. When system submissions are made to the SDL, the submitter has the option to activate a checkbox to indicate if the system includes localization information. When the checkbox is selected, NIST may activate spatial localization generation via the ActEV CLI experiment-init command and then use the merge-chunks to retrieve the spatial localizations as defined in Appendix D.2. Spatial localizations should NOT be produced via any other method (e.g., the ActEV command's merge-chunk's -o option).

> **Note:** The system activity file is structurally similar to the reference activity instance file described in Section 4 above. The reference annotation is used for for scoring and system training (therefore "processingReport" and "presenceConf" are NOT required) while system output contains the prediction output of the system (therefore "processingReport" and "presenceConf" are required).

Table 4: JSON format example for system output


```
{
  "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": "fail",
        "message": "ffmpeg exited with non-zero error code"
      }
    },
    "siteSpecific": {
    }
  },
  "activities": [
    {
      "activity": "person_closes_facility_door",
      "activityID": 1,
      "presenceConf": 0.832,
      "localization": {
        "2018-03-14.07-30-04.07-35-04.school.G336.avi": {
          "1": 1,
          "112": 0
        }
      },
      "objects": [
        {
          "objectType": "person",
          "objectID": 1,
          "localization": {
            "2018-03-14.07-30-04.07-35-04.school.G336.avi": {
              "10": { "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: A required array; enumerating the video 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 required dictionary; reporting success or failures forprocessing 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; the system can store additional information. The structure and content of this dictionary has no restrictions beyond the fact that it needs to be a syntactically correct JSON dictionary.
- activities: An array of annotated activity instances. Each instance is a dictionary with the following fields:
    - activity: The name (e.g. "person_talks_to_person") 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. 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 Spatio-Temporal 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 are 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 x-position, y-position, width and height respectively. Unlike the frame numbering system for a frame state signal, pixel coordinates are zero-based. Therefore, the top left pixel of the image is the pixel location (0,0) for the 'x' and 'y' values respectively. 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. For DIVA Performers, see "Appendix D" for detailed requirements and implementation information.

5.2. Validation of System Output File

The ActEV Scorer software package[2] 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. 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
```

The system output file can also be validated against the 'actev_sdl_v2_schema.json' (https://github.com/usnistgov/ActEV_Scorer/blob/master/lib/protocols/actev_sdl_v2_schema.json) using the command from the ActEV Evaluation CLI:

```
% jsonschema -i foo.json actev_sdl_v2_schema.json
```

6. 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.

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

The primary measure of performance will be the normalized, partial Area Under the DET (Detection Error Tradeoff) Curve 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:

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

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

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

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

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 6.5 for additional details)

Implementation notes:

- For $P_{miss}(x)$ where $\underline{x}$ is greater than the largest achieved T_{fa}, then $P_{miss}(x)$ is taken to be the minimum achieved $P_{miss}(x)$.
- When computing the first $P_{miss}(x)$ for the highest *presenceConf* value, $P_{miss}(x)$ will equal 1.0 between T_{fa}=0 and the first T_{fa} achieved be the system.
- 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 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.

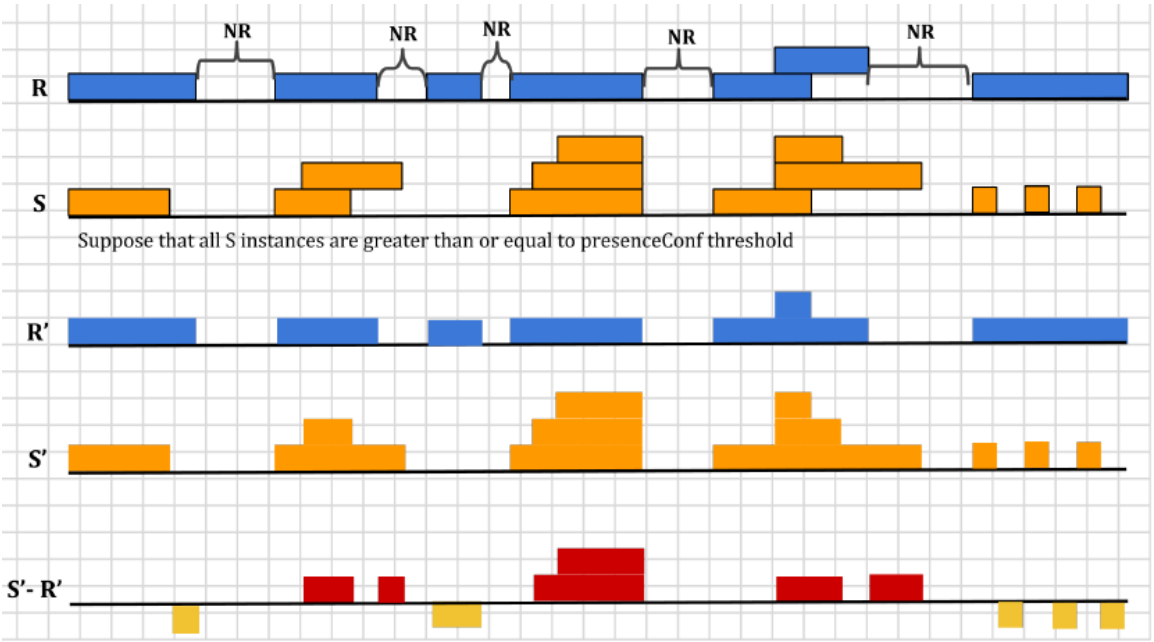

Figure 1: Pictorial depiction of T_{fa} calculation

(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 occur 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. 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 [11].

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 (Ns):

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

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.

6.3. 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.

6.4. 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.

6.5. 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[3]. 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
$K(I_{R_i}, I_{S_j}) = \{\emptyset \; if \; Activity\ (I_{S_j}) != Activity\ (I_{R_i})$

[3] 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.

when I_{R_i}>= 1 sec, $\emptyset$ if $Intersection(I_{R_i}, I_{S_j}) < 1$ sec,
when $I_{R_i} < 1$ sec, $\emptyset$ if $Intersection(I_{R_i}, I_{S_j}) <$ 50% of I_{R_i} $time$
$1 + AP_{con}(I_{S_j})$, $otherwise$}

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.

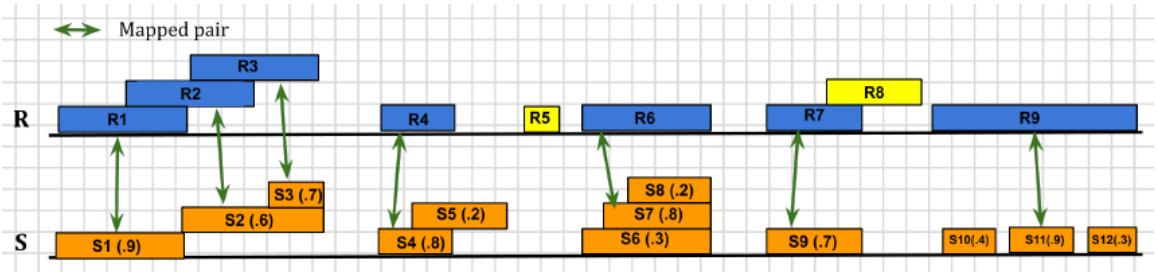

Figure 2: Pictorial depiction of activity instance alignment and P_{miss} calculation
(In *S*, the first number indicates instance id and the second indicates *presenceConf* score. For example, S1 (.9) represents the instance S1 with corresponding confidence score 0.9. Green arrows indicate aligned instances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

6.6. 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)

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 (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 CLI for the ActEV SDL evaluation, please visit the "Algorithm Submission" tab on the 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

APPENDIX D: CLI SPATIO-TEMPORAL LOCALIZATION REQUIREMENTS

DIVA performer systems are now required to include Spatio-Temporal Localization (STL) as a system output. This feature is being implemented to initiate research on future methods for quantitative performance/accuracy. To facilitate collection of STL data for evaluation, performers will update systems and will select which submissions include STL outputs via the CLI and SDL submission controls. Performers must select STL output as often as possible.

D.1 Spatio-Temporal Localization Output Specification

Teams will produce STL outputs as an 'activities' JSON as documented in Section 6.1 System Output File for Activity Detection Tasks. The STLs will be contained in the 'objects' element within each activity instance. The 'objects' element is an array supporting multiple items being localized. See "Section D.5: STL Validation and Examples" below for pointers to the schema, validation code, and example files. Refer to Section 6.1 above for the authoritative definition of the bounding box components (e.g., pixel origin, values, etc.).

Spatial bounding box region semantics are up to the system. The 'objectType' element, which is a string fill, can either match the Kitware annotations (person, vehicle, object) or be defined by the performer as either a general type or something specific. The documentation provided with the system must describe what the objects/bounding boxes mean for users.

Temporally varying bounding box (BBox) characteristics are up to the system but must be consistent within a single system. The bounding boxes will be rendered verbatim (per Section D.4) based solely on the values in the object's localization element in the JSON. Supported examples of the time-varying nature of the BBoxes are as follows. Examples of each can be found in Section D.5.

- *Single fixed BBox for the temporal span* - where a system produces one BBox per instance and that BBox is persistent for the duration for the activity instance.

- *Frame-spanned, varying BBoxes for a temporal span* - where the shape, size, and location of the BBoxes change throughout the duration of the activity instance.

Presently, there is no representation beyond a bounding box such as a hotspot. If another form of localization is needed, please contact NIST.

D.2 CLI Implementation and SDL Submission Controls

A STL output JSON is expected to be large and may increase computation time. While the program will require systems to always have the capability to produce localizations, the production and return of the STLs will be USER controlled. When selected, STLs are returned and all detected activity instances must be accompanied by STLs. Accordingly, there will be two major changes to the ActEV CLI and SDL evaluation.

First, the SDL system submission form will include a selection mechanism to indicate if the run should or should not produce STLs. [Note: There is a similar mechanism for a system to output a detection proposal list. While proposal generation is not required of the performers, the output mechanism was harmonized with the localization output mechanism and is documented below.]

Second, three command line programs will have their signatures changed:

- **actev experiment-init**: There will be two new options to communicate whether or not STLs and/or proposals will be collected from later steps. These are independent options. The options are: *--prepare-localization-outputs* and *--prepare-proposal-outputs* for STLs and proposals respectively. It is up to the implementation to react to the flag.
- **actev merge-chunks**: There will be two new optional arguments specifying the filename to write the outputs to. The options are '*--localization FILE*' and '*--proposals FILE*' for STLs and proposals respectively. These are independent options. If the option is used, an output file is expected. It is an error condition to use the merge_chunks option without the matching experiment_init option.
- **actev validate-execution**: There will be a new option *'--localization FILE*' to specify the STL output file and validate the file against the JSON Schema. See Appendix A for validation details. The proposals file will not be validated.

D.3 Evaluation of Spatial Localizations

NIST will validate the spatial localization output against a JSON Schema as defined in Section D.5. At present, the method of STL quantitative performance/accuracy assessments are to-be-determined but they may include assessment of all STLs. All STL localization measures are secondary to the system's overall Activity Detection performance and $P_{miss}@T_{fa} = 0.02$ remains the primary metric. STL measures will not be reported on the public SDL Leaderboard. NIST will share the measurements with IARPA and the teams.

Measurement options include, but are not limited to, the following. Like all ActEV metrics, measures will be computed separately by activity and then averaged over activities for an overall measure.

- $P_{miss}@T_{fa} = 0.02$ using localization accuracy as an additional correctness constraint.
- Precision of Detection of the top 100 highest presenceConfidence value instances. (aka, *precision @ 100*)

D.4 Visual Inspection

Kitware and NIST will generate instance clips with overlaid Reference and System spatial localization output for evaluation team use.

D.5 STL Validation and Examples

The STL output can be validated with the 'CLI_localization_schema.json' found in the ActEV Evaluation CLI validate_execution [12]. There are two ways to validate a STL output. NIST will use Option 2 in the SDL pipeline.

- Option1: Separate Command line.
  ```
  % python3
  diva_evaluation_cli/bin/private_src/implementation/validate_execution/validate_localization.py /path/to/your/localization/file.json
  ```

- Option 2: Via the ActEV Evaluation CLI (assuming the CLI is installed).
  ```
  % actev validate-execution [...] --localization
  /path/to/your/localization/file.json
  ```

Per Section D.1, the file format specifying STLs can be represented using three different time-varying methods and multiple objects. Presented below are examples:

- A single fixed BBox for the temporal span: 'single_bbox.json'
- Frame-spanned, frame-varying BBox for a temporal span: 'frame_spanning_varying_bbox.json'
- Multiple objects, single fixed BBox for a temporal span: 'multi_single_bbox.json'

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 Transactions on Pattern Analysis and Machine Intelligence, 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/
[11] ActEV Evaluation CLI: https://gitlab.kitware.com/actev/diva_evaluation_cli/
[12] ActEV Evaluation CLI validate_execution: https://gitlab.kitware.com/actev/diva_evaluation_cli/-/tree/development/diva_evaluation_cli/bin/private_src/implementation/validate_execution/

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.