ActEV Self-Reported Leaderboard (SRL) Challenge Draft Evaluation Plan

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

Date: March 04, 2022

ActEV Team, NIST

Contact: actev-nist@nist.gov

Updates

CVPR'22 ActivityNet ActEV Self-Reported Leaderboard (SRL) Schedule

- NIST releases CVPR'22 ActivityNet ActEV SRL test dataset (This is the same data name and data as used for the HADCV'22 workshop (ActEV SRL challenge))
- ActEV SRL Challenge Opens: March 15, 2022
- Deadline for ActEV SRL Challenge results submission: May 20, 2022
- Invite the top two teams on the ActEV SRL Challenge for CVPR'22 ActivityNet workshop : June 1, 2022

TRECVID 2022 Workshop: ActEV Self-Reported Leaderboard (SRL) Schedule

- NIST will release TRECVID'22 ActEV SRL test dataset: May 15, 2022
- ActEV SRL Challenge Opens: June 1, 2022
- Deadline for ActEV SRL results submission: October 20, 2022: 4:00 PM EST
- All teams are invited based on the participation to TRECVID 2022 Workshop: December 11, 2022

Old Updates

- Oct. 26, 2021
    - Added AOD as a primary task and redefined the AOD and AD tasks
    - Changed Section 6 Performance Metrics
    - Added Appendix D Alignment procedure
    - Removed the use of object bounding box presenceConf from the frame kernel function in Section "AOD Spatial Object Detection" Section because systems do not produce that information.

Table of Contents

1. Overview

The Activities in Extended Video (ActEV) series of evaluations is 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 to the running of the ActEV 2021 Sequestered Data Leaderboard (SDL) evaluations (see details at https://actev.nist.gov/sdl).

Currently under the CVPR'22 ActivityNet Guest Task, we are running the ActEV Self-Reported Leaderboard (SRL) Challenge which is based on the MEVA Known Facility (KF) datasets. The large-scale MEVA dataset is designed for activity detection in multi-camera environments. It was created on the Intelligence Advanced Research Projects Activity (IARPA) Deep Intermodal Video Analytics (DIVA) program to support DIVA performers and the broader research community. The ActEV Self-Reported Leaderboard (SRL) Challenge provides a new test set to participants to run activity detection systems on their own hardware platforms and submit their system outputs to the evaluation server for scoring. The TRECVID'22 ActEV task will also be running the ActEV Self-Reported Leaderboard (SRL) Challenge with an updated test dataset.

You can download the public MEVA KF test set from (https://mevadata.org) as described in Section 3 below. We also provide annotations for 160 hours of MEVA data, and instructions on how to make and share activity annotations are at https://mevadata.org/#service.

The ActEV Self-Reported Leaderboard (SRL) Challenge will report system performance scores on a public leaderboard on this website (https://actev.nist.gov/srl). Details of the performance metrics for Activity and Object Detection (AOD) and Activity Detection (AD) are described in Sections 6 and below.

The remainder of this document is organized as follows. Section 2 defines the evaluation tasks and conditions and Section 3 describes the data resources, system inputs and outputs are given Section 4 through 5, respectively. Section 6 defines the performance metrics for both AOD and AD. The detailed descriptions for Submission Instructions, Transmitting Submissions, Schema, data download, and alignment procedure are found in appendices.

2. Evaluation Task and Conditions

2.1. Task Definition

In the ActEV SRL evaluation, there are two tasks for systems; a primary task is Activity and Object Detection (AOD) and a secondary task is Activity Detection (AD)

Task1: The AOD task, given the predefined activity classes, the objective is to automatically detect the presence of the target activity and temporally/spatially localize all instances of the activity. This task requires spatio-temporal localization of objects involved in the activity (as one bounding box per frame that encompasses people, vehicles, and other objects) in the correspondence instance pairs. For a system-identified activity instance to be evaluated as correct, the activity class must be correct and the temporal/spatial overlap must fall within a minimal requirement. The evaluation tool, ActEV_Scorer, transforms the localization bounding boxes of both the system and reference files on the fly so that developers have the flexibility to spatially localize objects or the single encompassing box. See Sections 2.5 and 6.4

Task2: The AD task, the objective is to automatically detect the presence of the target activity and temporally localize all instances. This task does not require spatio-temporal localization of objects. For a system-identified activity instance to be evaluated as correct, the activity class must be correct and the temporal overlap must fall within a minimal requirement.

2.2. Conditions

The ActEV Self-Reported Leaderboard (SRL) Challenge 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 Self-Reported Leaderboard (SRL) challenge is a take-home evaluation; participants download ActEV SRL testset, run their activity detection algorithms on the test set using their own hardware platforms, and then submit their system output to the evaluation server for scoring results.

2.4. Protocol and Rules

During the ActEV SRL evaluation, you can create a maximum of four systems and submit a maximum of two results per day and a maximum of 50 results in total.

The challenge participants agree not to probe the test videos via manual/human means such as looking at the videos to produce the activity type and timing information from prior, during and after the evaluation.

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. On the Fly Bounding Box Localizations

The spatio-temporal localization requirements for the AOD task is specified based on encompassing, frame-varying bounding boxes. For example, if the activity is 'person-talks-to-person', the bounding box would encompass both people.

However, the reference annotation localizations have separate bounding boxes per person (or object) and developers have varying styles of bounding box identification. Therefore, both the system and reference annotations are transformed on the fly during the scoring. See Section 6.2 for the scoring commands to use.

3. Data Resources

The ActEV SRL evaluation is based on the Known Facilities (KF) data from the Multiview Extended Video with Activities (MEVA) dataset. 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) SRL test dataset.

The KF 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 (http://mevadata.org).

There are four locations of data pertaining to the MEVA data resources and the evaluation. The sections below document how to use the data for the CVPR'22 ActivityNet ActEV SRL and TRECVID'22 ActEV evaluations. The CVPR'22 ActivityNet ActEV SRL test dataset is the same as the one used for WACV'22 HADCV workshop ActEV SRL challenge.

- http://mevadata.org - general information about MEVA.
- AWS Video Data Bucket - The AWS bucket contains the video data for download.
- https://gitlab.kitware.com/meva/meva-data-repo - The GIT repo for public annotations.
- https://gitlab.kitware.com/actev/actev-data-repo - The GIT repo for files pertaining to CVPR'22 ActivityNet ActEV SRL and TRECVID'22 ActEV evaluations.

3.1. 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 http://mevadata.org.

The training video can be found in the AWS Video Data Bucket within the directories: *drops-123-r13*, *examples*, *mutc-3d-model*, *uav-drop-01*, and *updates-r13*. (NOTE: directory *drop-4-hadcv22* is NOT a training resource).

160 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 ActivityNet ActEV SRL and TRECVID'22 ActEV (ActEV SRL) evaluation-related materials. The evaluations make use of multiple data sets. This repo is a nexus point between the evaluations and the utilized data sets. The repo consists of partition definitions (e.g., train, validation, or test) to be used for the evaluations.

3.2. Self-Reported Leaderboard Test Dataset

The CVPR'22 ActivityNet ActEV SRL test dataset is a 16-hour collection of videos which only consists of Electro-Optics (EO) camera modalities from public cameras. The test dataset is the same as the one used for WACV'22 HADCV workshop ActEV SRL challenge. The TRECVID'22 ActEV SRL test dataset will be updated and released on May 15th.

The test data for evaluation can be found in the AWS Video Data Bucket within the *drop-4-hadcv22* directory

The evaluation activity-index and file-index JSONs can be found in the actev-data-repo in the dataset partition directory 'partitions/HADCV22-Test-20211010'.

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:

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

The table below shows the names of the *20 Known Activities* for ActEV SRL evaluations.

ActEV SRL Known Activity Names

person_closes_vehicle_door	person_reads_document
person_enters_scene_through_structure	person_sits_down
person_enters_vehicle	person_stands_up
person_exits_scene_through_structure	person_talks_to_person
person_exits_vehicle	person_texts_on_phone
person_interacts_with_laptop	person_transfers_object
person_opens_facility_door	vehicle_starts
person_opens_vehicle_door	vehicle_stops
person_picks_up_object	vehicle_turns_left
person_puts_down_object	vehicle_turns_right

4. 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 mevadata.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) and found in the actev-data-repo in the dataset partition directory 'partitions/HADCV22-Test-20211010'.

5. System Output

In this section, the system output format is defined. The ActEV Scorer software package[1] 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.

5.1. System Output File for Activity Detection Tasks

The system output file should be a JSON file that includes a list of videos processed by the system, an optional execution report of file processing success and failures, and a collection of activity instance records with temporal localizations and spatial localizations of objects.

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": [
    "2018-03-07.16-50-00.16-55-00.hospital.G479.avi"
  ],
  "processingReport": {
    "fileStatuses": {
      "2018-03-07.16-50-00.16-55-00.hospital.G479.avi": {
        "status": "success",
        "message": "hello world"
      }
    },
    "siteSpecific": {}
  },
  "activities": [
    {
      "activity": "Talking",
      "activityID": 1,
      "localization": {
        "2018-03-07.16-50-00.16-55-00.hospital.G479.avi": {
          "1": 1,
```

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

```
        "20": 0,
        "100": 1,
        "112": 0,
      }
     },
     "objects": [
      {
        "objectType": "person",
        "objectID": 1,
        "localization": {
          "2018-03-07.16-50-00.16-55-00.hospital.G479.avi": {
            “1": { “boundingBox”: {"x":10, "y":30,"w":50,"h":20}},
            “20": {},
            “100": { “boundingBox”: {"x":10, "y":30,"w":50, "h":20}},
            “104": { “boundingBox”: {"x":60, "y":60,"w":50, "h":20}},
            “108": { “boundingBox”: {"x":30, "y":90,"w":50, "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 “executionReporting” dictionary below can be used to report anomalies.
- 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. Each Frame State Signal (for a video) has keys representing a frame number and the value being 1 (the activity instance is present) and 0 (otherwise) within the given 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.

    - objects: An array of objects annotated with respect to the activity instance. The objects are 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 objects. The value must be unique within a single JSON file.
        - Localization: The spatio-temporal localization of the objects referred to by the record encoded as a dictionary of Frame State Signals indexed by the video file id for which the objects are witnessed. Each Frame State Signal (for a given video) has keys representing a frame number and the value is a dictionary describing one spatial localization that encompasses all the objects involved in the activities. The spatial dictionary 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.
- processingReport: An optional 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. All files need not be present.
        - <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.

5.2. Validation of Activity Detection System Output

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

```
% python3 ActEV_Scorer.py ActEV_SRL_V2 -V -s SYSTEM.json  -a
activity-index.json -f file-index.json
```

6. Performance Metrics

The primary measure for the ActEV SRL evaluation is the probability of missed detection (P_{miss}) at a specified Rate of False Alarm (R_{FA}), namely ($P_{miss}@R_{FA}$). The secondary measures are a normalized, partial area under the DET curve ($nAUDC$) and average Mean Average Precision (mAP) over a set of IoU thresholds.

The technologies sought for the ActEV SRL leaderboard evaluation are expected to report activities that visibly occur in a single-camera video for a user to review. Systems will identify each activity instance by specifying the activity label, the video file, the frame span of the activity, the spatio-temporal localization of the objects involved in the activity, and the *presenceConf* value indicating the system's 'confidence score' that the activity is present. The presenceConf value is used for all metric computations to calculate performance across the full range of operating points (in presenceConf space).

6.1. Computation of Pmiss and Rfa

The performance measures computation can be summarized into the four steps; 1) one-to-one instance alignment, 2) confusion matrix computation, 3) metrics summarization, and 4) aggregation and visualization.

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

The measure requires a one-to-one correspondence between pairs of reference and system output activity instances. The following descriptions are a modified version of the TRECVID 2017: Surveillance Event Detection [1] framework.

For a target activity, multiple instances can occur within the same duration. For example, instances R_2 and R_3 occur in different locations within the same duration in a video as shown in Figure 1.

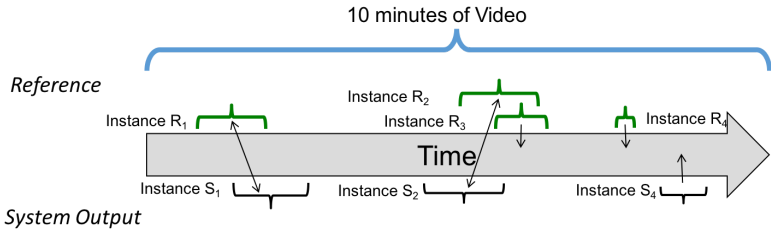

Figure 1. The instance alignment of the reference and system output

To compare the instances of the system output and the reference annotations, the scoring tool will first find the corresponding instances between reference and system output. For an optimal one-to-one instance mapping, the tool utilizes the Hungarian solution to the Bipartite Graph matching problem [2] using different “kernel” functions (“*K*” functions defined in Appendix D) to determine mappable reference and system instances by measuring the congruence between the reference annotations and the system output. For the AOD task, the kernel function requires both temporal and spatio-temporal congruence while the AD task requires temporal congruence ignoring spatial object information (see the details in Appendix D).

Step 2: Confusion Matrix Computation

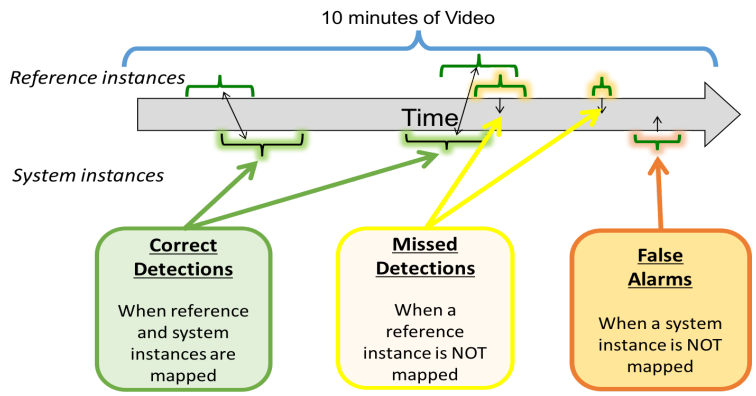

Figure 2. Confusion matrix computation

Figure 2 illustrates the alignment between the reference and system instances and the resulting confusion matrix labels. The matrix is defined as:

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

Step 3: Performance Metrics Summarization

Following the calculation of the detection confusion matrix, the next step is to summarize the performance metrics. Each instance counts are accumulated by comparing the presenceConf score to a certain threshold; instances with a score greater than or equal to the threshold is interpreted as a decision of "yes", indicating that the system's belief is that the activity instance is a target activity; instances with a score less than the threshold is interpreted as a decision of "no", indicating that the system's belief is that the instance is not a target activity. For activity instance occurrence, a probability of missed detections (P_{miss}) and a rate of false alarms (R_{FA}) at a given threshold τ can be computed:

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

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

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

Step 4: Aggregation and Visualization

P_{miss} and R_{FA} values are calculated for each activity and visualized using **Detection Error Tradeoff (DET) curve.** The DET curve is used as one of the graphical performance analysis tools. The y-axis is the probability of missed detections. The x -axis is the rate of false alarms. Martin et at al. [3] provide detailed information about DET curves for detection system evaluation. Figure 3 illustrates a DET curve. P_{miss} and R_{FA} values can be aggregated over all the activity classes and summarized as to single value.

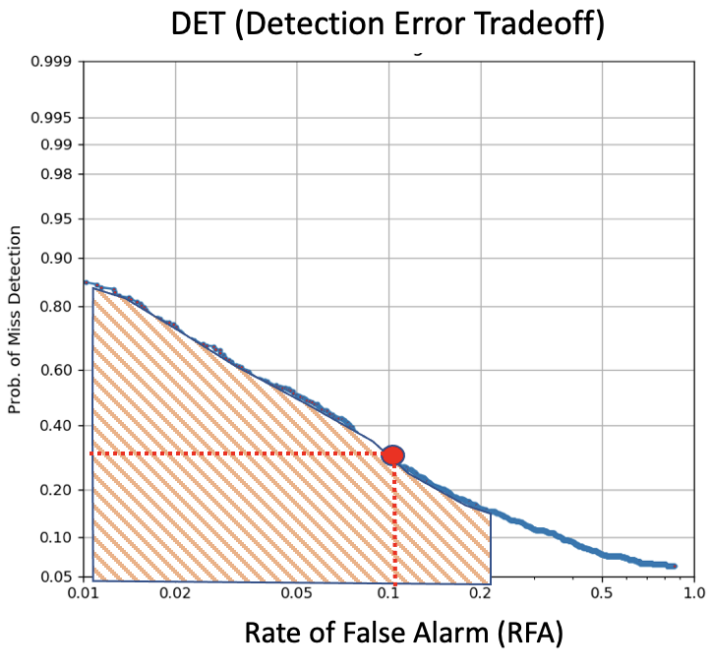

Figure 3. The Detection Error Tradeoff (DET) with P_{Miss}@0.1R_{FA} and nAUDC@0.2R_{FA}

Probability of Missed Detection at Fixed Rate of False Alarm (P_{miss}**@**R_{FA}**)** is used to report a point for the Probability of Missed Detection corresponding to a specified Rate of False Alarm; for ActEV SRL, we will report, P_{miss}**@**$0.1R_{FA}$, the point marked in red in Figure 3.

6.2. Computation of nAUDC

Normalized, partial Area Under the DET Curve ($nAUDC$) from 0 to a fixed, Rate of False Alarm (R_{fa}) value *a*, denoted $nAUDC_a$. The partial area under the 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:

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

where x is integrated over the set of R_{fa} values.

For ActEV SRL, we will report nAUDC from 0 to 0.2 R_{fa}, area marked in orange in Figure 3.

6.3. Computation of mAP

For the evaluation, we will report mAP and the average mAP. To calculate the mAP for Activity Detection, first the Interpolated Average Precision (AP) is calculated as the evaluation metric for the submitted results for each activity. Then, the AP is averaged over all the activity categories to get the mAP value. To determine if a detection is a true positive, we compare the temporal intersection over union (tIoU) with a ground truth segment, and check whether or not it is greater than or equal to a given threshold (e.g. tIoU > 0.5). The average mAP metric is defined as the mean of all mAP values computed with tIoU thresholds between 0.5 and 0.95 (inclusive) with a step size of 0.05.

6.4. ActEV_Scorer Command Line

The ActEV_Scorer command supports many evaluation "protocols" (*PROTO*) for the ActEV evaluations. The general form of the command line is:

```
% ActEV_Scorer.py PROTO -s system.json -r reference.json -a
activity-index.json -f file-index.json -o output-folder -F -v
```

To 'validate' system output against a particular protocol's output schema, add '-V'

For the AD Task, use the "SRL_AD_V1" protocol.

For the AOD Task, use the "SRL_AOD_V1" protocol and add the options:

- '--transformations single_bbox_per_frame' to enable the bounding box localization transformation to a single bounding box per frame.
- '--rewrite .transformed' to re-rewrite the modified system output for manual inspection.

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

APPENDIX

Appendix A: Submission Instructions

System output and documentation submission to NIST for subsequent scoring must be made using the protocol, consisting of three steps: (1) preparing a system description and self-validating system outputs, (2) packaging system outputs and system descriptions, and (3) transmitting the data to NIST.

The packaging and file naming conventions for ActEV evaluation series rely on **Submission Identifiers** (SubID) to organize and identify the system output files and system description for each evaluation task/condition. Since SubIDs may be used in multiple contexts, some fields contain default values. The following EBNF (Extended Backus-Naur Form) describes the SubID structure with several elements:

<SubID> ::= <SYS>_<VERSION>_[OPTIONAL]

<SYS> is the SysID or system ID. No underscores are allowed in the system ID. The team is allowed to have the two submissions only; primary and secondary respectively. It should begin with 'p-' for the one primary system (i.e., your best system) or with 's-' for the one secondary system. It should then be followed by an identifier for the system (only alphanumeric characters allowed, no spaces). For example, this string could be "p-baseline" or "s-deepSpatioTemporal". This field is intended to differentiate between runs for the same evaluation condition. Therefore, a different SysID should be used for runs where any changes were made to a system.

<VERSION> should be an integer starting at 1, with values greater than 1 indicating multiple runs of the same experiment/system.

[OPTIONAL] is any additional string that may be desired, e.g. to differentiate between tasks. This will not be used by NIST and is not required. If left blank, the underscore after <VERSION> should be omitted.

As an example, if the team is submitting on the AD task using their third version of the primary baseline system, the SubID could be:

p-baseline_3_AD

A.1 System Descriptions

Documenting each system is vital to interpreting evaluation results. As such, each submitted system, determined by unique experiment identifiers, must be accompanied by a system description with the following information.

Section 1 Submission Identifier(s)

List all the submission IDs for which system outputs were submitted. Submission IDs are described in further detail above.

Section 2 System Description

A brief technical description of your system.

Section 3 System Hardware Description and Runtime Computation

Describe the computing hardware setup(s) and report the number of CPU and GPU cores. A hardware setup is the aggregate of all computational components used.

Report salient runtime statistics including: wall clock time to process the index file, resident memory size of the index, etc.

Section 5 Training Data and Knowledge Sources

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

Section 6 System References

List pertinent references, if any.

A.2 Packaging Submissions

Using the SubID, all system output submissions must be formatted according to the following directory structure:

<SubID>/

<SubID>.txt	The system information file, described in Appendix A-a
<SubID>.json	The system output file, described in Section 5.1

As an example, if the earlier team is submitting, their directory would be:

p-baseline_3_AD/

p-baseline_3_AD.txt

p-baseline_3_AD.json

A.3 Transmitting Submissions

To prepare your submissions, first create the previously described file/directory structure. Then, use the command-line example to make a compress the TAR or ZIP file:

$ tar -zcvf SubID.tgz SubID/ e.g., tar -zcvf p-baseline_3_AD.tgz p-baseline_3_AD/

$ zip -r SubID.zip SubID/ e.g., zip -r p-baseline_3_AD.zip p-baseline_3_AD/

To submit the output to NIST, log into https://actev.nist.gov website and make a submission per the instructions https://actev.nist.gov/uassets/instructions.pdf.

Please submit your files in time for us to deal with any transmission errors that might occur well before the due date if possible. Note that submissions received after the stated due dates for any reason will be marked late.

Appendix B: SCHEMAS

B.1 JSON Schema for system output file

Please refer to the ActEV_Scorer software package (same for the ActEV evaluations) (https://github.com/usnistgov/ActEV_Scorer) for the most up-to-date schemas, found in “lib/protocols”.

B.2 Scoring Server

The team will submit their system output in the Json file format described earlier to an online web based evaluation server application at NIST. The initial creator of the team on the scoring server will have control over who can submit system outputs on behalf of the team using a username and a password. The evaluation server will validate the file format and then compute scores. The server will be available for teams to test the submission process.

Appendix C: Data Download

C.1 ActEV SRL Dataset

The ActEV Self-Reported Leaderboard (SRL) Challenge is based on the Multiview Extended Video with Activities (MEVA) Known Facility (KF) dataset. The MEVA KF data was collected at the Muscatatuck Urban Training Center (MUTC) with a team of over 100 actors performing in various scenarios. The MEVA KF dataset has two parts: (1) the public training and development data and (2) ActEV SRL test dataset (available Sep 10th, 2021).

The MEVA KF 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.

C.2 Training and Development Data

In December 2019, the public MEVA KF dataset was released with 328 hours of ground-camera data and 4.2 hours of Unmanned Arial Vehicle video. 160 hours of the ground camera video was 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.

C.3 ActEV SRL Test dataset

The CVPR'22 ActivityNet ActEV SRL Test dataset has been released. The test dataset is the same as the one used for WACV'22 HADCV workshop ActEV SRL challenge. The TRECVID'22 ActEV SRL test dataset will be updated and released on May 15th.

There are four locations of data pertaining to the MEVA data resources and the evaluation. The sections below document how to obtain and use the data for the HADCV evaluation.

- http://mevadata.org - general information about MEVA.
- MEVA AWS Video Data Bucket - The AWS bucket contains the video data for download.
- https://gitlab.kitware.com/meva/meva-data-repo - The GIT repo for public annotations.
- https://gitlab.kitware.com/actev/actev-data-repo - The GIT repo for files pertaining to CVPR'22 ActivityNet ActEV SRL and TRECVID'22 ActEV SRL evaluations. This repo is the distribution mechanism for the CVPR'22 ActivityNet and TRECVID'22 (ActEV SRL) evaluation-related materials. The evaluations make

use of multiple data sets. This repo is a nexus point between the evaluations and the utilized data sets. The repo consists of partition definitions (e.g., train, validation, or test) to be used for the evaluations.

C.4 Data Download

You can download the public MEVA video for free from the mevadata.org website (http://mevadata.org/) by completing these steps:

C.5 CVPR'22 ActivityNet ActEV SRL Test data

- 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
    - Follow the steps in the top-level README.
    - Download the HADCV22 SRL test data collection into ./partitions/HADCV22-Test-20211010 using the command:

        % python scripts/actev-corpora-maint.py --regex ".*drop-4-hadcv22.*" --operation download

C.6 TRECVID'22 ActEV SRL Test data

The TRECVID'22 ActEV SRL test dataset will be updated and released on May 15th

- More information is coming soon.

C.6 MEVA Training/Development data

- 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
    - Download the training data collection found in the MEVA AWS Video Data Bucket within the directories: drops-123-r13, examples, mutc-3d-model, uav-drop-01, and updates-r13. (NOTE: directory drop-4-hadcv22 is **NOT** a training resource).

APPENDIX D: ACTIVITY INSTANCE MAPPING PROCEDURE

A single system instance cannot be counted as correct for multiple reference instances[2]. In order to optimally determine which system instances are correct and which reference instances are missed, the evaluation code performs an optimal, reference-to-system instance mapping that minimizes the measured P_{miss}. The mapping is computed using the Hungarian algorithm solving the Bipartite Graph matching problem [2], which reduces the computational complexity and arrives at an optimal solution using a 'kernel' function that measures the fitness of mapping a single system/reference instance pair. In our implementation, the kernel function does the following.

1. Enforces a minimum consistency between the reference and system instances determining a potential map. These minimums are expressed as thresholds on various congruences such as temporal overlap and spatio-temporal overlap. The various tasks use different instance attributes. AD uses temporal overlap; AOD adds spatio-temporal overlap.
2. The alignment prefers aligning pairs with higher presenceConf detections to minimize the measured error as well as produce DET curves with thresholding instances based on presenceConf values.

The balance of this section covers the definitions of the mapping kernel functions beginning with the temporal activity detection kernel (for the AD task) and then the spatio-temporal activity and object detection kernel (for the AOD task) which extends the kernel to objects.

D.1 AD:TEMPORAL ACTIVITY DETECTION

[2] 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.

The mapping kernel function for the AD task uses three bits of information:

- The system and reference activity type must match. If the requirement is not met, ∅ is returned indicating a mapping is not permitted.
- The temporal congruence between a system and reference instance meets a minimum. If the requirement is not met, ∅ is returned indicating a mapping is not permitted.
- The preference of choosing a particular mapping takes into account the relative temporal congruence and the relative value of the system instance's presenceConf value.

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

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

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

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

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

where,

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

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}

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

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

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

$E_{IoU} = 1.0e-8$; a constant to weight overlap ratio congruence

$E_{AP} = 1.0e-6$; a constant to weight activity presence confidence score congruence

$AP_{con}(I_{S_j})$: 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; ∅ indicates that the pairs of reference and system output instances are not mappable due to either missed detections or false alarms, otherwise the pairs of instances have a score for potential match.

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

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

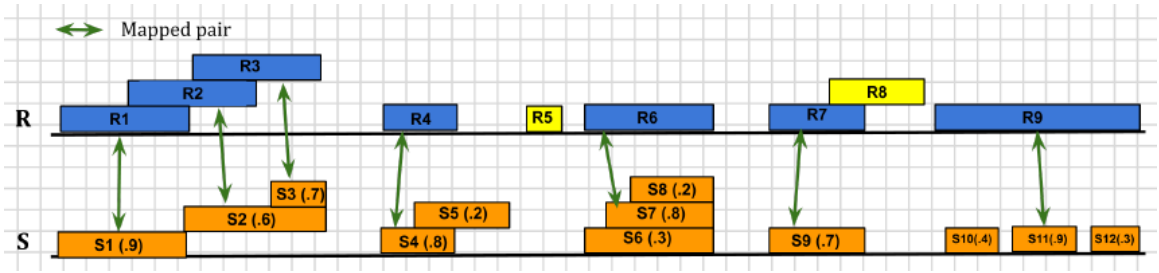

Figure 4: Pictorial depiction of activity instance alignment and P_{miss} calculation

(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 4, 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 4, 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).

D.2 AOD: Spatio-Temporal Activity Detection

For ActEV systems, activities and objects localized for activity instances can occur at any point in the video's 3D volume and the annotated objects (typically a person, vehicle, other objects) are a subset of the total inventory visible in the scene. Therefore, for the AOD task, the evaluation code must determine if the temporal ranges are sufficiently congruent as well as determine if the annotated reference objects match the objects localized by the system. The mapping kernel function for AOD scoring uses a two-level mapping strategy whereby there is an activity instance mapping (that is similar to AD) that incorporates an instance-aggregated congruence of the reference and system objects that is estimated via an object-level application of the Hungarian algorithm.

The following information is used for the instance alignment:

- The system and reference activity types must match. If the requirement is not met, ∅ is returned indicating a mapping is not permitted.
- The temporal congruence between a system and reference instance meets a minimum. If the requirement is not met, ∅ is returned indicating a mapping is not permitted.
- (For objects) The Normalized Multiple Object Detection Error, (n_MODE defined below) meets a minimum threshold. ∅ is returned indicating a mapping is not permitted.
- The preference of choosing a particular mapping takes into account the relative temporal congruence, the relative value of the system instance's presenceConf value, and the relative spatio-temporal object congruence.

The AOD Kernel Function *K* is defined as below. See Section D.1 for the previous variable definitions.

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

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

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

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

$$\emptyset \; if \; O_c(I_{R_i}, I_{S_j}) < \Delta O_c$$

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

ΔO_c=0.2 (final value TBD; will calibrate based on current system performance); constant setting the minimum spatio-temporal overlap of reference and system objects
$E_O = 10^{-10}$; a constant to weight object detection congruence
$O_c(I_{R_i}, I_{S_j})$= 1 - MODE; the object detection congruence function between a reference and system output instance defined in Section D.2.1.

The additional constants E_O sets the relative importance of the object detection information source.

D.2.1 AOD Spatial Object Detection

For AOD spatial object detection, we employ the N_MODE (Normalized Multiple Object Detection Error) metrics described in [4][5] and the presentation below is a simplification tailored to the ActEV evaluation approach. N_MODE evaluates the relative number of false positives and missed detections for objects per activity instance. **Note that these metrics are applied only to the frames and objects in correspondence instance pairs.**

The metric includes the object alignment using frame-level object mappings between reference bounding boxes and system output bounding boxes using the Hungarian algorithm (see the paper [4] for the detailed descriptions). For the object alignment procedure, the following new kernel function K_O is used. See Section D.1 for the variable definitions.

$K_O(B_{R_i}, \emptyset) = 0$: the kernel value for an unmapped reference object

$K_O(\emptyset, B_{S_j}) = -1$: the kernel value for an unmapped system object

$K_O(B_{R_i}, B_{S_j})$= {$\emptyset$ if $ObjectType$ (B_{S_j}) ! = $ObjectType$ (B_{R_i})

$$\emptyset \; if \; Spatial_{IoU}(B_{R_i}, B_{S_j}) <= \Delta Spatial_{IoU}$$

$$1 + Spatial_{IoU}(B_{R_i}, B_{S_j}), \quad otherwise\}$$

B_{R_i}: the i^{th} reference bounding box of the objects encompassed

B_{S_j}: the j^{th} system output bounding box of the objects encompassed

$Spatial_{IoU}(B_{R_i}, B_{S_j})$: $Intersection(B_{R_i}, B_{S_j})$ over $Union(B_{R_i}, B_{S_j})$ in a spatial domain

$\Delta Spatial_{IoU}$= 0.3; the fixed spatial IoU threshold

The confusion matrix for each frame *t* using the output of the mapping. Note that we only consider frames within the intersection of the reference and system temporal localizations.

- Correct Detection (CD_t): the count of reference and system output bounding boxes that are mapped to each other for frame t.
- Missed Detection (MD_t): the count of reference bounding boxes not mapped to a system output bounding box.
- False Alarm (FA_t): the count of system bounding boxes not mapped to a reference object.
- Correct Rejection: this metric is not calculated in this evaluation.

Using the frame-based object confusion matrix, N_MODE (Normalized Multiple Object Detection Error) is computed aggregating over frames and is defined as:

$$N\_MODE = \frac{\sum_{t=1}^{N_{frames}} C_{MD} * MD_t + C_{FA} * FA_t}{\sum_{t=1}^{N_{frames}} N_R^t}$$

where:

MD_t: the object missed detections in frame t

FA_t: the object false alarms in frame t

C_{MD}=1: the cost function for missed detections
C_{FA}=1: the cost function for false alarms
N_R^t: the number of reference objects in frame t
N_{frames}: the number of frames in the sequence for the instance

N_MODE is the object detection performance for an activity instance, however since the choice of system instances to evaluate are a function of the threshold on the activity instance *presenceConf* value, the evaluation code aggregates *N_MODE* to summarize object detection performance for specific RFA thresholds reporting activity MEAN N_MODE @ RFA=X and also the mean over activities.

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
[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.