Draft ActEV Sequestered Data Leaderboard (SDL) Evaluation Plan

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

Date: 2019-08-12

ActEV Team, NIST

Contact: actev-nist@nist.gov

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 for forensic and real-time alerting applications. ActEV began with the Summer 2018 Blind and Leaderboard evaluations and has currently progressed to the running of two concurrent evaluations: 1) the TRECVID 2019 ActEV self-reported leaderboard based on the VIRAT V1 and V2 datasets [9] with 18 target activities and, 2) an independent evaluation called the ActEV Sequestered Data Leaderboard (ActEV SDL) based on the Multiview Extended Video (MEVA) dataset [10] with 37 target activities.

The ActEV SDL evaluation provides a mechanism for evaluating activity detection algorithms on challenging extended duration video. Activities in extended video are dispersed temporally and spatially requiring algorithms to detect and localize activities under a variety of collection conditions. Multiple activities may occur simultaneously in the same scene, while extended periods may contain no activities. Participants are invited to submit their runnable activity detection software using an ActEV Command Line Interface (CLI) submission. NIST will then evaluate system performance on sequestered data using NIST hardware and results will be posted to a public leaderboard. See Appendix B for pointers to the ActEV CLI.

Challenge participants will develop activity detection and temporal localization algorithms for 37 activities that are to be found in extended videos and video streams. These videos contain significant spans without any activities and intervals with potentially multiple concurrent activities.

The ActEV SDL evaluation is based on the Multiview Extended Video with Activities (MEVA) dataset. Participants are encouraged to annotate the data for the 37 activities as described at mevadata.org. For ActEV participants, the MEVA dataset is available for download without a fee. Please click on the data tab of the ActEV SDL website (actev.nist.gov/sdl) for information on how to download the data.

For this evaluation plan, an activity is defined to be “one or more people performing a specified movement or interacting with an object or group of objects”. Detailed activity definitions are in the ActEV Annotation Definitions for MEVA Data document [7]. Each activity is formally defined by four elements:

Element	Meaning	Example Definition
Activity Name	A mnemonic handle for the activity	Open Trunk
Activity Description	Textual description of the activity	A person opening a trunk
Begin time rule definition	The specification of what determines the beginning time of the activity	The activity begins when the trunk lid starts to move
End time rule definition	The specification of what determines the ending time of the activity	The activity ends when the trunk lid has stopped moving

2. Tasks and Conditions

2.1. Tasks

In the ActEV SDL evaluation, there is one Activity Detection (AD) task for detecting and localizing activities.

For the AD task, given a target activity, a system automatically detects and temporally localizes all instances of the activity. For a system-identified activity instance to be evaluated as correct, the type of activity must be correct, and the temporal overlap must fall within a minimal requirement as described in Section 6.

2.2. Conditions

The ActEV SDL evaluation will focus on the forensic analysis that processes the full corpus prior to returning a list of detected activity instances.

2.3. Evaluation Type

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

2.4. Protocol and Rules

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

System runtime must be less than or equal to 1 x the data length [see Appendix D].

2.5. Required Evaluation Condition

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

ActEV SDL Independent	**Required**
Task	Activity Detection
Target Application	Forensic Systems
Evaluation Type	Sequestered Evaluation
Submission	See the details in Appendix A for Submission Instructions
Dataset	MEVA

3. Data Resources

The ActEV SDL evaluation is based on the Multiview Extended Video with Activities (MEVA) dataset. The MEVA dataset has two parts, the sequestered test dataset, and the MEVA Known Facility Release #1 (KF1) data that contains approximately 185 hours of video collected at the Muscatatuck Urban Training Center with a team of over 100 actors performing in various scenarios. For ActEV participants, the MEVA KF1 dataset is available for download without a fee. Please click on the data tab on the website (actev.nist.gov/sdl) for information on how to download the data [see Appendix C].

The table below provides a list of activities for the ActEV SDL evaluation. 37 target activities are used in the ActEV SDL evaluation. The detailed activity definitions are in the ActEV Annotation Definitions for MEVA Data document (https://gitlab.kitware.com/meva/meva-data-repo/blob/master/documents/MEVA-Annotation-Definitions.pdf). ActEV participants are encouraged to annotate the MEVA KF1 dataset for the 37 activities as described at mevadata.org.

person_closes_facility_door
person_closes_vehicle_door
Closing_Trunk
person_enters_through_structure
person_enters_vehicle
person_exits_through_structure
person_exits_vehicle
person_loads_vehicle
Open_Trunk
person_opens_facility_door
person_opens_vehicle_door
Transport_HeavyCarry
person_unloads_vehicle
vehicle_turning_left
vehicle_turning_right
vehicle_u_turn
Riding
Talking
specialized_talking_phone
specialized_texting_phone
person_sitting_down
person_standing_up
person_reading_document
object_transfer
person_picks_up_object
person_sets_down_object
hand_interaction
person_person_embrace
person_purchasing
person_laptop_interaction
vehicle_stopping
vehicle_starting
vehicle_reversing
vehicle_picks_up_person
vehicle_drops_off_person

```
abandon_package
theft
```

4. System Input

Along with the source video files, the subset of video files to process for evaluation will be specified in a provided file index JSON file [8]. Systems will also be provided an activity index JSON file, which lists the activities to be detected by the system.

4.1. File Index

The file index JSON file lists the video source files to be processed by the system. Note that systems need only process the selected frames (as specified by the “selected” property). An example, along with an explanation of the fields is included below.

```
{
   "2018-03-07.16-50-00.16-55-00.hospital.G479.avi": {
        "framerate": 30,
        “camera_id”: “G479”,
        “camera_type”: “EO”,
        “begin_time”: “2018-03-07.16-50-00”
        “end_time”: “2018-03-07.16-55-00”
        "selected": {
           "1": 1,
           "20941": 0
        }
   },
   "2018-03-07.16-50-06.16-55-06.school.G336.avi": {
        "framerate": 30,
        “camera_id”: “G336”,
        “camera_type”: “EO”,
        “begin_time”: “2018-03-07.16-50-00”
        “end_time”: “2018-03-07.16-55-00”
        "selected": {
          "11": 1,
          "201": 0,
          "300": 1,
          "20656": 0
        }
    }
}
```

- <file>:
  - framerate: number of frames per second of video
  - camera_id: G336; the id of the camera
  - camera_type: ‘EO’ | ‘EO-NIR’ | ‘IR’; for Electro Optical, Electro Optical-Near Infrared, or Infrared respectively
  - begin_time: the beginning date/time stamp of the recording
  - end_time: the ending date/time stamp of the recording
  - selected: The on/off signal designating the evaluated portion of <file>
    - <framenumber>: 1 or 0, indicating whether or not the system will be evaluated for the given frame. Note that records are only added here when the value changes. For example, in the above sample, frames 1 through 20940 in file “2018-03-07.16-50-00.16-55-00.hospital.G479.avi” are selected for processing/scoring. The default signal value is 0 (not-selected), and the frame index begins at 1, so for file “2018-03-07.16-50-00.16-55-06.hospital.G336.avi”, frames 1 through 10 are not selected. Also note that the signal must be turned off at some point after it’s been turned on, as the duration of the signal is needed for scoring.

4.2. Activity Index

The activity index JSON file lists the activities to be detected by the system. An example, along with an explanation of the fields is included below.

```
{
 "Closing": {},
 "Closing_Trunk": {},
 "Entering": {},
 "Exiting": {},
 "Loading": {}
}
```

- <activity>: A collection of properties for the given <activity>. For SDL, no further properties are specified for activity.

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. Challenge 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, along with a collection of activity instance records with spatio-temporal localization information (depending on the task). A notional system output file is included inline below, followed by a description of each field.

```
{
 "filesProcessed": [
  "2018-03-07.16-50-00.16-55-00.hospital.G479.avi"
 ],
 "activities": [
  {
   "activity": "Talking",
   "activityID": 1,
   "presenceConf": 0.89,
   "localization": {
    "2018-03-07.16-50-00.16-55-00.hospital.G479.avi": {
     "1": 1,
     "20": 0
    }
   }
  }
 ]
}
```

- filesProcessed: the list of video source files processed by the system
- activities: the list of detected activities; each detected activity is a record with the following fields:
    - activity: (e.g. “Talking”)

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

- activityID: a unique identifier for the activity detection, should be unique within the list of activity detections for all video source files processed (i.e. within a single system output JSON file)
- presenceConf: The score is any real number that indicates the strength of the possibility (e.g., confidence) that the activity instance has been identified. The scale of the presence confidence score is arbitrary but should be consistent across all testing trials, with larger values indicating greater chance that the instance has been detected. Those scores are used to generate the detection error tradeoff (DET) curve.
- localization (temporal): The temporal localization of the detected activity for each file
    - <file>: The on/off signal temporally localizing the activity detection within the given <file>
        - <framenumber>: 1 or 0, indicating whether the activity is present or not, respectively. Systems only need to report when the signal changes (not necessarily every frame)

5.2. Validation of Activity Detection System Output

The system output file will be validated against a JSON Schema (see, ActEV Scorer: https://github.com/usnistgov/ActEV_Scorer). Further semantic checks may be performed prior to scoring by the scoring software. E.g. checking that the video list provided in the system output is congruent with the list of files provided to the system for evaluation.

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

```
% ActEV_Scorer.py ActEV19_AD_V2 -V -s SYSTEM.json  -a activity-index.json
-f file-index.json
```

6. Activity Detection Metrics

The technologies sought for the ActEV SDL leaderboard evaluation are expected to report activities that visibly occur in a single-camera video by identifying the video file, the frame span of the activity, and the *presenceConf* value indicating the system’s ‘confidence score’ that the activity is present.

The primary measure of performance will be the normalized, partial Area Under the DET Curve ($nAUDC$) from 0 to a fixed, Time-based False Alarm (T_{fa}) value a, denoted $nAUDC_a$.

The partial area under DET curve is computed separately for each activity over all videos in the test collection and then is normalized to the range [0, 1] by dividing by the maximum partial area a. $nAUDC_a = 0$ is a perfect score. The $nAUDC_a$ is defined as:

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

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

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

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

N_{frames} : The duration (frame-based) of the video
NR : Non-Reference duration. The duration of the video without the target activity occurring
S'_i : the total count of system instances for frame i
R'_i : the total count of reference instances for frame i
T_{fa} : The time-based false alarm value(see Section 6.1 for additional details)
$N_{md}(x)$: the number of missed detections at the presenceConf threshold that result in $T_{fa} = x$
$N_{TrueInstance}$: the number of true instances in the sequence of reference
$P_{miss}(x)$: The probability of missed detections (instance-based) value for $T_{fa} = x$ value (see Section 6.2 for additional details)

Implementation notes:

- If T_{fa} never reaches a, the system's minimum value of P_{miss} is used through a
- If the T_{fa} value occurs between two *presenceConf* values, a linearly interpolated value for *presenceConf* is used

6.1. Computation of Time-Based False Alarm

Time-based false alarm (T_{fa}) is the fraction of non-activity instance time (in the reference) for which there is a system that falsely identified an instance. All system instances, regardless of overlap with references instances, are included in this calculation and overlapping system instances contribute double or more (if there are more than two) to the false alarm time. Also note, temporally fragmented system detections that occur during non-activity time do not increase T_{fa} unless they overlap temporally.

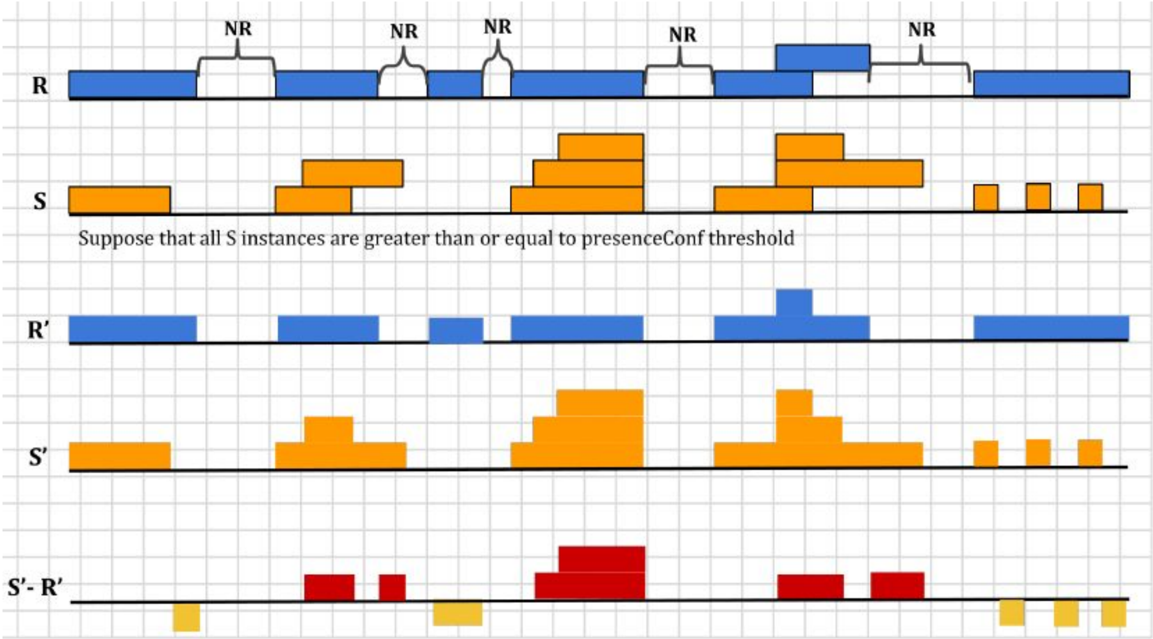
NR
NR
NR
NR
NR
R
S
Suppose that all S instances are greater than or equal to presenceConf threshold
R'
S'
S'- R'

Figure 1: Pictorial depiction of T_{fa} calculation

(R is the reference instances and S is the system instances. R' is the histogram of the count of reference instances and S' is the histogram of the count of system instances for the target activity.)

In Equation (2), first the non-reference duration (NR) of the video where no target activities occurs is computed by constructing a time signal composed of the complement of the union of the reference instances durations. As depicted in the Figure above, R' and S' are histograms of count instances across frames (N_{frames}) for the reference instances (R) and system instances (S), respectively. R' and S' both have N_{frames} bins, thus R'_i is the value of the i^{th} bin of R' and S'_i is the value of the i^{th} bin of S'. S'_i is the total count of system instances in frame i and R'_i is the total count of reference instances in frame i.

False alarm time is computed by summing over positive difference of $S'_i - R'_i$ (shown in red in the figure above); that is the duration of falsely detected system instances. This value is normalized by the non-reference duration of the video to provide the T_{fa} value in Equation (2).

6.2. Alignment used in Computation of Probability of Missed Detection

A missed detection is a reference activity instance that the system did not detect. The Probability of Missed Detection (P_{miss}) is the fraction of reference instances not detected by the system.

As an instance-measure of performance, a single system instance cannot be counted as correct for multiple reference instances[2]. In order to optimally determine which instances are missed, and thereby minimize the measured P_{miss}, the evaluation code performs a reference-to-system instance alignment algorithm that minimizes the measured P_{miss} factoring the *presenceConf* values so that a single alignment also minimizes the $nAUDC$.

While the mapping procedure is one-to-one, system instances not mapped are ignored, effectively allowing a 1-to-many alignment because many system instances that overlap with a reference instance are not penalized in the P_{miss} calculation. However, all system instances can contribute to the T_{fa} calculation.

The alignment is computed between the reference instances and system detected instances using the Hungarian algorithm to the Bipartite Graph matching problem [2], which reduces the computational complexity and arrives at an optimal solution such that:

1. Correctly detected activity instances must meet a minimum temporal overlap with a single reference instance.
2. System instances can only account for one reference instance (otherwise, a single, full video duration system instance would be aligned to N reference instances).
3. The alignment prefers aligning higher presenceConf detections to minimize the measured error.

In 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

[2] For instance, if there are two abandon_bag activity instances that occur at the same time but in separate regions of the video and there was a single detection by the system, one of the reference instances was missed.

correspondence procedure for instances is performed for a single target activity (A_i) at a time.

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

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

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

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

when $I_{S_j} < 1$ sec, $\varnothing$ *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; $\varnothing$ indicates that the pairs of reference and system output instances are not mappable due to either missed detections or false alarms, otherwise the pairs of instances have a score for potential match.

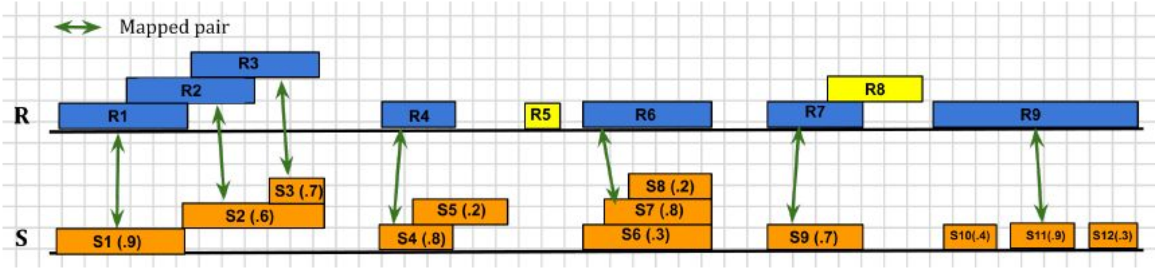

Figure 2: Pictorial depiction of activity instance alignment and P_{miss} calculation (In *S*, the first number indicates instance id and the second indicates *presenceConf* score. For example, S1 (.9) represents the instance S1 with corresponding confidence score 0.9. Green arrows indicate aligned instances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

In the example of Figure 2, for the case of reference instances {R1, R2, R3} and system instances {S1, S2, S3}, either R2 or R3 can be considered as a missed detection depending on the way reference instances are mapped to system instances. To minimize P_{miss} for such cases, the alignment algorithm is used to determine one-to-one correspondence as to {R1, S1}, {R2, S2}, and {R3, S3}. It also identifies system instance S7 as a better match to reference instance R6 factoring the *presenceConf* values.

In Equation (3), $N_{TrueInstance}$ represents the number of true instances in the sequence of reference and N_{md} is the number of nonaligned reference instances that are missed by the system. In Figure 2, suppose that the *presenceConf* threshold is greater than or equal to 0.5. Thereby, $N_{TrueInstance}$ is 9 and N_{md} is 2 (marked in yellow).

6.3. ACTEV_Scoring Command Line

The command to score a system using the ActEV_Scorer is:

```
% ActEV_Scorer.py ActEV19_AD -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 ActEV19_AD -s system-output.json -a activity-index.json
  -f file-index.json -F -v -V
```

APPENDIX

Appendix A: NIST Independent Evaluation Infrastructure Specification

Hardware specification:

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

Appendix B: ActEV Command Line Interface for Software Delivery

The challenge participants will deliver their algorithms that are compatible with the ActEV Command Line Interface (ActEV CLI) protocol to NIST. The CLI documentation prescribes the steps to install the software/algorithm from a web-downloadable URL and run the algorithm on the Independent Evaluation Infrastructure. For more information on the ActEV Command Line Interface (ActEV CLI) for the ActEV SDL evaluation, please visit the "Algorithm Submission" tab on the ActEV SDL website (https://actev.nist.gov/sdl).

Appendix C: Data Download

To download the 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 updated a previously downloaded repo with 'git pull'. Note: this is the same repo as used from VIRAT.
    - Clone: git clone https://gitlab.kitware.com/actev/actev-data-repo.git
    - Update: cd "Your_Directory_For_actev-data-repo"; git pull
- Add MEVA download credentials:
    - Change your working directory the top-level of the repo.

        -
          ```
          cd "Your_Directory_For_actev-data-repo"
          ```
    - Follow the steps in the top-level README.
    - For Step 2 in the download instructions, use this command to add your access credentials. (Please do not email this command!)

        -
          ```
          python ./scripts/actev-corpora-maint.py --operation summary
          --corpus MEVA --add_credential '{"corpus": "MEVA", "urls":
          {"https://mig.nist.gov/datasets/MEVA": {"type": "file_store",
          "user": "MEVA", "password": "???????"}}}'
          ```

Appendix D: System Runtime Speed

System runtime must be less than or equal to $1x$ the video duration and is calculated as follows:

$SyTime =$ *ActEV-design-chunks + ActEV-experiment-init + ActEV-pre-process-chunk + ActEV-process-chunk + ActEV-post-process-chunk + ActEV-merge-chunk + ActEV-experiment-cleanup*

RTFactor = SyTime / Video_duration

RTFactor should be less than or equal to 1.

References

[1] TRECVID 2017 Evaluation for Surveillance Event Detection, https://www.nist.gov/itl/iad/mig/trecvid-2017-evaluation-surveillance-event-detection

[2] J. Munkres, “Algorithms for the assignment and transportation problems,” Journal of the Society of Industrial and Applied Mathematics, vol. 5, no. 1, pp. 32–38, 1957

[3] Martin, A., Doddington, G., Kamm, T., Ordowski, M., Przybocki, M., “The DET Curve in Assessment of Detection Task Performance”, Eurospeech 1997, pp 1895-1898.

[4] K. Bernardin and R. Stiefelhagen, “Evaluating multiple object tracking performance: The clear mot metrics,” EURASIP Journal on Image and Video Processing, vol. #, 2008.

[5] R.Kasturi et al.,“Framework for performance evaluation of face, text, and vehicle detection and tracking in video: Data, metrics, and protocol,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 31, no. 2, pp. 319–336, Feb. 2009.

[6] Kitware DIVA Annotation Guidelines, Version 1.0 November 6, 2017.

[7] ActEV Annotation Definitions for MEVA Data document: https://gitlab.kitware.com/meva/meva-data-repo/blob/master/documents/MEVA-Annotation-Definitions.pdf

[8] MEVA Annotation JSON document: https://gitlab.kitware.com/meva/meva-data-repo/blob/master/documents/MEVA_Annotation_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.