Draft of ActEV 2018 Evaluation Plan

Date: 2018-10-26

ActEV Team

NIST

Table of Contents

1. Introduction

The Activities in Extended Video (ActEV) 2018-2019 challenge will evaluate technologies that detect activities in the real-world case of multiply collected, extended videos such as surveillance video where there are no shot boundaries to segment the video and the video continues to be recorded for an indefinite period of time. In the future, this plan will feed into a continuing evaluation series that will measure many aspects of the produced technologies including, but not limited to, performance accuracy, speed of computations, response time of the detection, and among others via several evaluation tasks.

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". Activities are determined during annotations and defined in the data selections below. Each activity is formally defined by five 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
Required object type list	The list of objects systems are expected to identify for the activity	ObjectType(s): Person, Vehicle

For the purposes of ActEV evaluation, we define a **World Activity Instance** (WAI) – as an activity performed by person(s) at a specific point in time and location in the real world and an **Activity Observation** (AO) – a sensor recording of a world activity instance.

To illustrate the difference, consider two scenarios:

- Scenario 1: one person picks up a box which is recorded by 3 cameras. There is 1 WAI with 3 AOs of that WAI.
- Scenario 2: a person (P_a) picks up a box by a door and another person (P_b) picks up a box by the car. There are 2 partial overlapping cameras: one camera records both people, the second camera records just P_b. There are two WAIs: (1) for P_a there are 2 AOs, (2) for P_b there is 1 AO.

The remainder of this evaluation plan covers resources, task definitions, task conditions, file formats for system input and output, evaluation metrics, scoring procedures, and protocols for submitting results.

Any questions or comments concerning the ActEV evaluation plan should be sent to ActEV-nist@nist.gov.

2. Tasks and Conditions

2.1. Tasks

In the ActEV evaluation, there will be multiple tasks for detecting and localizing primitive and complex activities and tracking multiple objects in a multi-camera video streaming environment. For ActEV evaluation, systems may leverage multiple cameras but evaluation will be performed within single camera view and at the Activity Observation level.

For this evaluation, we defined three tasks: 1) Activity Detection (AD), 2) Activity and Object Detection (AOD), and 3) Activity and Object Detection and Tracking (AODT). Each task can be completed independently. **The ActEV 1.A evaluation focused on the tasks AD and AOD only while the 1.B evaluation includes the tasks AD, AOD, and AODT.**

2.1.1. Activity Detection (AD)

For the Activity Detection 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.

2.1.2. Activity and Object Detection (AOD)

For the Activity and Object Detection task, given a target activity, a system detects and temporally localizes all instances of the activity and spatially detects/localizes the people and/or objects associated with the target activity. For a system-identified instance to be scored as correct, it must meet the temporal overlap criteria for the AD task and in addition meet the spatial overlap of the identified objects during the activity instance as described in Section 6.

2.1.3. Activity Object Detection and Tracking (AODT)

For the Activity Object Detection and Tracking task, given a target activity, a system detects and temporally localizes all instances of the activity, spatio-temporally detects/localizes the people and/or objects associated with the target activity, and properly assigns IDs the objects play in the activity. For a system-identified instance to be scored as correct, it must meet the temporal overlap criteria and spatio-temporal overlap of the objects for the AOD task and correctly assign the IDs to the objects as prescribed in the activity definition as described in Section 6.

2.2. Conditions

The technology developed for the evaluation is expected to be applied to both forensic analysis (applications that process vast collections for repeated investigation) and alerting (applications that process many video streams with detection occurring within defined latency during collection). Both applications require quick

processing of video, be it by parallelization or minimal computation. For alerting scenarios, additionally, a system must be able to detect if an activity is occurring during the onset of the activity even though the entire video (or video stream), nor the end of the activity, has been processed. For this the evaluation, we define **detection latency** to be the time from the onset of the event to the last frame processed by the system before being able to declare the activity is occurring.

With forensic and alerting applications in mind, the evaluation differentiates two types of systems: forensic systems and low latency systems. The ActEV 2018 evaluation will focus on the forensic systems only.

2.2.1. Forensic Systems

For a forensic system, the system processes the full corpus prior to returning a list of detected activity instances.

2.2.2. Low Latency Systems

For a low latency system, the system detects the presence of the target activity instance and reports its detection latency at the point in time (or frame) when the system determined the instance occurred in a video streaming environment.

2.2.3. Reference Temporal Segmentation

For the reference temporal segmentation evaluation condition, systems are given the temporal localization of each activity instance in the video. This condition is a controlled evaluation condition to test the systems' ability to classify activity instances alone rather than performing both segmentation and instance classification.

2.3. Evaluation Type

For the ActEV evaluation, there are the two evaluation types; self-reported evaluation and sequestered evaluation.

2.3.1. Self-reported evaluation

For self-reported evaluation, the performers should run their software on their systems and configurations and submit the system output defined by this document (see Section 5) to the NIST Scoring Server.

2.3.2. Independent/Sequestered evaluation

For independent/sequestered evaluation, the performers should submit their runnable system to NIST using the forthcoming Evaluation Container Submission Instructions. NIST will evaluate system performance on sequestered data using NIST hardware--see the details in Appendix C for the hardware infrastructure.

2.4. Protocol and Rules

The performers can train their systems or tune parameters using any data complying with applicable laws and regulations. All data used for training is expected to be made available by performers after the initial evaluation cycle where the data is used. In the event that external limitations preclude sharing such data with others, performers are still permitted to use the data, but they must inform NIST that they are using such data, and provide appropriate detail regarding the type of data used and the limitations on distribution.

The performers 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 to the evaluation period until permitted by NIST.

All machine learning or statistical analysis algorithms must complete training, model selection, and tuning prior to running on the test data. This rule does not preclude online learning/adaptation during test data processing so long as the adaptation information is not reused for subsequent runs of the evaluation collection.

The only VIRAT data that may be used by the systems are the ActEV-provided training and validation sets, associated annotations, and any derivatives of those sets (e.g., additional annotations on those videos). All other VIRAT data and associated annotations may not be used by any of the systems for the ActEV evaluations.

For the reference temporal segmentation evaluation (when applicable), the performer must, to the extent possible, use the same underlying classifier for the evaluation. The provided segmentations are allowed to use for online learning/adaptation during test data processing.

2.5. Required Evaluation Condition

For ActEV 1.A evaluation, the conditions can be summarized as shown in Table below:

ActEV 1.A Evaluation	**Required**	**Optional**
Task	AD, AOD	
Conditions	Forensic Systems Reference Temporal Segmentation	
Evaluation Type	Self-reported Evaluation	
Submission	Primary (see the details in Appendix A for Submission Instructions)	Secondary
Data Sets	VIRAT-V1	

For ActEV 1.B evaluation, the conditions can be summarized as shown in Table below:

ActEV 1.B Evaluation	**Required**	**Optional**
Task	AD, AOD, AODT	
Conditions	Forensic Systems	
Evaluation Type	Self-reported Evaluation Independent Evaluation	
Submission	Primary (see the details in Appendix A for Submission Instructions)	Secondary
Data Sets **(Self-reported eval)**	VIRAT-V1 VIRAT-V2	

Data Sets (Independent Eval)	VIRAT-V1 VIRAT-V2 M1	

3. Data Resources

This document does not contain details on data used for the ActEV evaluations. Please note that development and evaluation data resources will be provided by Kitware Inc.

The table below provides a summary of the activity list and the required objects for the ActEV evaluations. The twelve target activities are used in the 1.A evaluation while nineteen target activities are used in both the ActEV Leaderboard and ActEV 1.B evaluations. The target objects for the ActEV evaluations (1.A, Leaderboard, and 1.B) are P = {Person} and V = {Construction_Vehicle, Vehicle}. The detailed definitions of the activities and its associated objects are found in Appendix D.

Table: List of activities per task and required objects

ActEV 1.A Eval	**ActEV LeaderBoard Eval**	**ActEV 1.B Eval**
Closing (P, V) or (P) Closing_trunk (P, V) Entering (P, V) or (P) Exiting (P, V) or (P) Loading (P, V) Open_Trunk (P, V) Opening (P, V) or (P) Transport_HeavyCarry (P, V) Unloading (P, V) Vehicle_turning_left (V) Vehicle_turning_right (V) Vehicle_u_turn (V)	Closing (P, V) or (P) Closing_trunk (P, V) Entering (P, V) or (P) Exiting (P, V) or (P) Loading (P, V) Open_Trunk (P, V) Opening (P, V) or (P) Transport_HeavyCarry (P, V) Unloading (P, V) Vehicle_turning_left (V) Vehicle_turning_right (V) Vehicle_u_turn (V) Interacts (P) Pull (P) Riding (P) Talking (P) activity_carrying (P) specialized_talking_phone (P) specialized_texting_phone (P)	Closing (P, V) or (P) Closing_trunk (P, V) Entering (P, V) or (P) Exiting (P, V) or (P) Loading (P, V) Open_Trunk (P, V) Opening (P, V) or (P) Transport_HeavyCarry (P, V) Unloading (P, V) Vehicle_turning_left (V) Vehicle_turning_right (V) Vehicle_u_turn (V) Interacts (P) Pull (P) Riding (P) Talking (P) activity_carrying (P) specialized_talking_phone (P) specialized_texting_phone (P)

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. Systems will also be provided with 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.

```
{
  "VIRAT_S_000000.mp4": {
      "framerate": 30,
      "selected": {
        "1": 1,
        "20941": 0
      }
  },
  "VIRAT_S_000001.mp4": {
      "framerate": 30,
      "selected": {
      "11": 1,
        "201": 0,
        "300": 1,
        "20656": 0
      }
  }
}
```

- <file>:
    - framerate: number of frames per second of video
    - 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 "VIRAT_S_000000.mp4" are selected for processing/scoring. The default signal value is 0 (not-selected), and the frame index begins at 1, so for file "VIRAT_S_000001.mp4", 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": {
    "objectTypes": [
       "Door",
        "Person",
```

```
        "Vehicle"
    ]
  },
  "Closing_Trunk": {
    "objectTypes": [
        "Person",
        "Vehicle"
    ]
  },
  "Entering": {
    "objectTypes": [
        "Door",
        "Person",
        "Vehicle"
    ]
  },
  "Exiting": {
    "objectTypes": [
        "Door",
        "Person",
        "Vehicle"
    ]
  },
  "Loading": {
    "objectTypes": [
        "Person",
        "Vehicle",
        "Prop"
    ]
  }
}
```

- <activity>: A collection of properties for the given <activity>
    - objectTypes: the set of objects to be detected by the system for the given activity

5. System Output

In this section, the types of system outputs are defined. The ActEV Score package[1] contains a submission checker that validates the submission in both the syntactic and semantic levels. Participants should check their submission prior to sending them to NIST. We will reject submissions that do not pass validation. The ActEV Scoring Primer document contains instructions for how to use the validator. NIST will provide the command line tools to validate submission files.

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

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. Regarding file naming conventions for submission, please refer to Appendix A.

Note that some fields may be optional depending on which task the system output is submitted for.

```
{
  "filesProcessed": [
    "VIRAT_S_000000.mp4"
  ],
  "activities": [
    {
      "activity": "Talking",
      "activityID": 1,
      "presenceConf": 0.89,
      "alertFrame": 20,
      "localization": {
        "VIRAT_S_000000.mp4": {
          "1": 1,
          "20": 0
        }
      },
      "objects": [
        {
          "objectType": "person",
          "objectID": 1,
          "localization": {
            "VIRAT_S_000000.mp4": {
               "1": { "presenceConf": 0.45, "boundingBox": { "x": 10,
"y": 30, "w": 50, "h": 20 } }
               "20": {}
            }
          }
        }
      ]
    }
  ]
}
```

- 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")
    - 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.
- alertFrame: the time point when the system determined the instance occurred; see the figure 1. The reported frame should be relative to the start of the video.
- localization (temporal): The temporal localization of the detected activity for each file
    - <file>: The on/off signal temporally localizating 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)
- objects: the list of constituent objects for the detected activity; each being a record the following fields, meant to represent the track of that object:
    - objectType: e.g. "Person"
    - objectID: A unique identifier for the detected object; should be unique within the activity instance (these uniqueness requirements may change as the evaluation tasks evolve).
    - localization (spatial): The spatial localization (track) of the object detection within the given <file>
        - <file>: The spatial localization signal of the object, as a bounding box and presence confidence score.
            - <framenumber>:
                - "boundingBox": bounding box record, using pixel coordinates [origin being the top-left corner pixel (x, y)], which spatially localizes the object for the given <framenumber>. Systems only need to report when the signal changes (not necessarily every frame)
                    - x
                    - y
                    - w: width
                    - h: height
                - "presenceConf": any real number that indicates the strength of the possibility (e.g., confidence) that an object of a given type exists at this spatial location. Your system should normalize these values across activity instances, as we wish to measure object detection performance over an aggregate of activity instances.

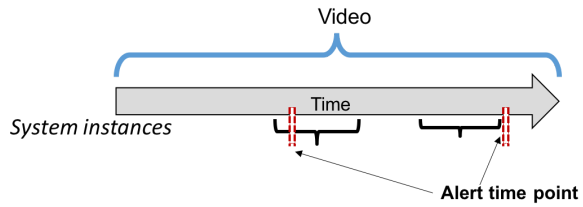

Figure 1. Example of system reported alert time

5.2. Validation of Activity Detection System Output

The system output file will be validated against a JSON Schema (see Appendix B), 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 teams for evaluation.

6. Activity Detection Metrics

The technologies sought for the ActEV evaluation are expected to report activities that occur in the ensemble of video(s) by identifying the camera(s) for which the activity is visible, reporting the time span of the activity, detecting the people and objects involved in the activity, specifying the type of each object in the activity, and tracking multiple objects in multiple cameras.

This ambitious list of system capabilities is organized as a set of increasingly demanding evaluation tasks where the definition of 'correct detection' requires more detailed system output specificity. For example, Activity Detection (AD) requires temporal accuracy, Activity and Object Detection (AOD) adds object detection and localization, and Activity Object Detection and Tracking (AODT) adds object type identification to the tracked objects.

There will be a two-pronged approach to evaluating system performance for the AD, AOD, and AODT tasks. The first prong, the primary measures, will compute system performance using the binary detection measure of the probability of missed detection (P_{miss}) at a fixed false alarm ($Rate_{FA}$). This measure will be applied to all tasks so that a degradation in performance between the AD, AOD, and AODT tasks are using a consistent set of measures for a given corpus.

The second prong, the secondary measures, will evaluate the quality of the description of the activity instance. The secondary measures vary depending on output produced by the system for a given task. For instance, the temporal localization measure is a secondary metric for the AD, AOD and AODT tasks while spatio-temporal localization of objects applies to the AOD and AODT tasks.

The following table identifies the performance questions answered by this protocol in terms of the primary and secondary measures for the AD, AOD, and AODT tasks.

Tasks	Primary Question/Metric	Secondary Question/Metric	Evaluated System Instance Content
AD, AOD, AODT	Can a system temporally detect instances of a target activity X? $P_{miss}@Rate_{FA} = X$	How accurate is the temporal localization of the detected instance? - Normalized Multiple Instance Detection Error (N-MIDE)	Activity, BeginFrame, EndFrame, Latency*, Score
AOD, AODT	Can a system temporally detect instances of activity X and detect the presence of objects (object type and bounding box) involved in the instance? $P_{miss}@Rate_{FA} = X$	How accurate is the spatial localization of objects involved in the activity? - Minimum Normalized Multiple Object Detection Error (minMODE) - $P_{miss}@Rate_{FA} = X$	Activity, BeginFrame, EndFrame, Latency*, Score, ObjectType ObjectBoundingBox
AODT	Can a system temporally detect instances of activity X and detect the presence of objects involved in the instance and assign the object identity over time? $P_{miss}@Rate_{FA} = X$	How accurate is the spatio-temporal localization of the tracked objects involved in the activity? - Minimum Multiple Objects Tracking Error (minMOTE) - $P_{miss}@Rate_{FA} = X$	Activity, BeginFrame, EndFrame, Latency*, Score, ObjectType, ObjectBoundingBox, ObjectIdentity

* if provided by a low-latency system

The following description is a scoring protocol for the primary/secondary performance measures for each task. Given reference and system output, in general, the scoring procedure can be divided into four distinctive steps: Alignment, Detection Confusion Matrix, Performance Metrics, and Result Visualization. Note that the secondary metrics are applied to corresponding instance pairs only after the alignment procedure.

The similar evaluation steps are used for each additional evaluation task, with the only differences being in the alignment step where additional system output instance content is taken into account and additional secondary performance measures are added.

6.1. Activity Detection Task

6.1.1. Activity Instance occurrence detection per activity (Primary Metric)

This metric evaluates performance on whether the system correctly detects the presence of the target activity instance. 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.

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

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 2.

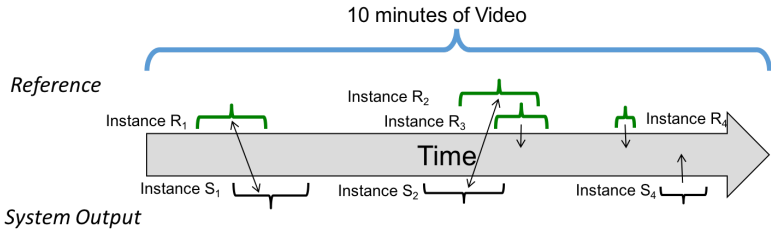

Figure 2. The instance alignment of the reference and system output

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], which reduces the computational complexity and arrives at an optimal solution.

In this approach, the reference instances are represented as one set of nodes and the system output instances are represented as one set of nodes. The mapping kernel function K below assumes that the one-to-one correspondence procedure for instances is performed for a single target activity (A_i) at a time.

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

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

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

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

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

where,

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

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}
$Union(I_{R_i}, I_{S_j})$: 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 [the value can be changed through evaluations]
$E_{IoU} = TBD$; a constant to weight overlap ratio congruence
$E_{AP} = TBD$; a constant to weight activity presence confidence score congruence
$AP_c(I_{S_j})$: 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.

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 to 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.

Step 2: Confusion Matrix Computation

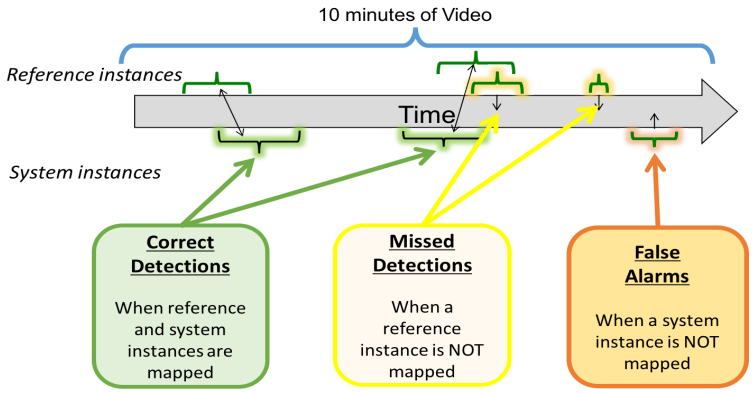

Figure 3. Confusion matrix computation

Figure 3, illustrates the confusion matrix calculation for activity instance occurrence and 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

Following the calculation of the detection confusion matrix, the next step is to summarize the performance metrics. Each trial's score will be converted to a decision by comparing the score to a certain threshold; a trial 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; a trial 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 and a rate of false alarms at a given threshold τ can be computed:

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

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

$P_{miss}(\tau)$: the probability of missed detections at the activity presence confidence score threshold τ.

$Rate_{FA}(\tau)$: 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: Result visualization

Detection Error Tradeoff (DET): 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 al. [3] provide detailed information about DET curves for detection system evaluation. Figure 5 illustrates a DET curve.

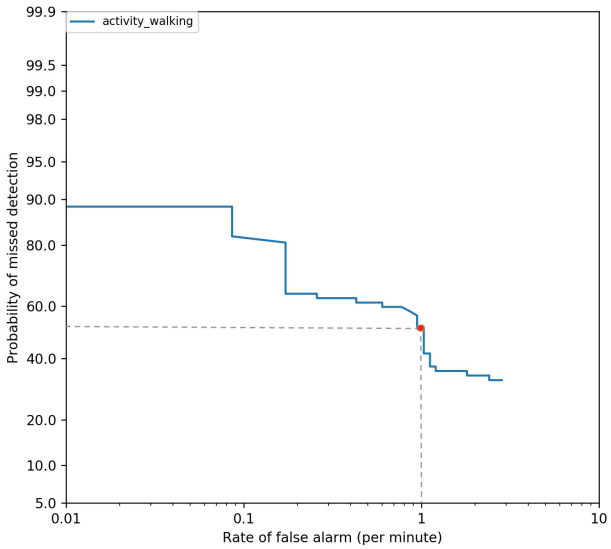

Figure 5. The Detection Error Tradeoff (DET) curve

Probability of Missed Detection at Fixed Rate of False Alarm (P_{Miss}@$Rate_{FA}$): Another graphical analysis used is to report a point for the Probability of Missed Detection on the DET corresponding to a specified Rate of False Alarm—see the point marked in red in Figure 5.

For ActEV, we will report P_{Miss} at many fixed $Rate_{FA}$ points including: 1, 0.2, 0.15, 0.1, 0.03, and 0.01 false alarms per minute. For Phase 1 (1.A and 1.B) evaluations, system performance will primarily be evaluated at two operating points; P_{Miss} at $Rate_{FA}$=0.15 for Class A activities and P_{Miss} at $Rate_{FA}$=1 for Class B activities. The activity classes are characterized by performance of system output and baseline output. For phase 1 evaluations, the Class A activities are considered to be easier to detect compared to the Class B activities.

6.1.2. Temporal Localization in Activity Instances (Secondary Metric)

This metric evaluates performance on how precisely activity instances have been detected for corresponding instance pairs. The following Figure 6 illustrates the confusion matrix computation.

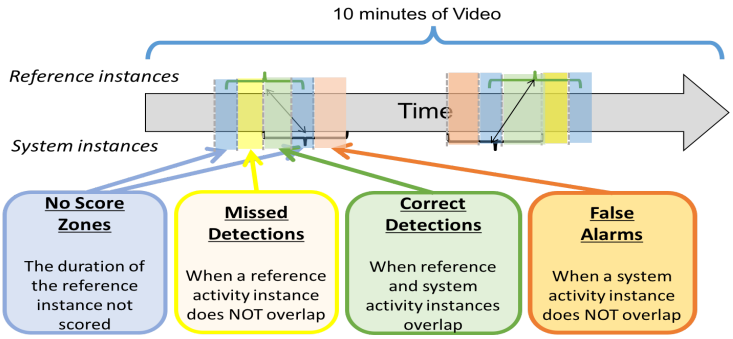

Figure 6. illustrates the confusion matrix computation

- Correct Detection (CD_I): the duration distance where reference and system output instances overlap. The example duration is shown in green.
- Missed Detection (MD_I): the duration distance where an instance in the reference does not overlap with an instance with same label in the system output. The example duration is shown in yellow.
- False Alarm (FA_I): the duration distance where an instance in the system output has no overlap with an instance with same label in the reference. The example duration is shown in orange.
- Correct Rejection: The reference indicates it is no instance duration, and the system output also does not detect it as an instance duration. This is not computable in this evaluation.
- No-Score (NS_I): Due to annotation error/ambiguity of beginning/ending time for the activity, we utilize a No Score Zone, that is the duration of the reference instance not scored. The distance marked in blue indicates the no score zone. Temporal regions indicated as occluded will be treated as no-score regions.

Thus, we compute the normalized multiple instance detection error (N_MIDE) as:

$$N\_MIDE = \frac{\sum_{I=1}^{N_{mapped}} \left(C_{MD} * \frac{MD_I}{MD_I + CD_I} + C_{FA} * \frac{FA_I}{Duration_V - (MD_I + CD_I + NS_I)}\right)}{N_{mapped}}$$

MD_I : the duration distance of missed detections in the instance I
FA_I : the duration distance of false alarms in the instance I
NS_I : the no-score distance in the instance I
C_{MD} : the cost parameter for weighting missed detections
C_{FA} : the cost parameter for weighting false alarms
N_{mapped} : the number of mapped instance pairs between reference and system output
$Duration_V$: the duration of the reference video V

Where C_{MD} and C_{FA} are the cost function for the missed detections and false alarms. C_{MD} and C_{FA} can be varied based on the specific application [4]. For example, the missed detections are more critical than false alarms, we can increase C_{MD} and reduce C_{FA}. In this evaluation, C_{MD} and C_{FA} are both equal to 1 and we use the minimum threshold τ of the activity presence confidence score to calculate the N_MIDE default value in ActEV-Scorer. In addition, multiple N_MIDE values are calculated at different operating points--for example, N_MIDE at Rate_{FA}=0.15 for Class A activities and N-MIDE at Rate_{FA}=1 for Class B.

6.2. Activity and Object Detection (AOD) Task

For this task, the metrics below evaluate system performance on 1) whether the system correctly detects the presence of the target activity and 2) whether the system correctly detects objects, in terms of the object type and bounding box, within a mapped instance pair.

6.2.1. Activity and Object Detection (Primary Metric)

For the AOD task, the primary metric stays the same as AD, P_{Miss}@Rate_{FA}, but the instance alignment step (see Step 1 above in Section 6.1.1) uses an addition term for object detection as an additional requirement for correctness (i.e., correspondence).

For activity instance alignment procedure for AOD task, using the same bipartite graph matching algorithm for reference and system alignment, the following kernel function K is used. See Section 6.1.1 for the previous variable definitions.

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

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

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

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

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

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

See Section 6.1.1 for the previous variable definitions
ΔO_c =0.0 (final value TBD; will calibrate based on current system performance); constant setting the minimum spatio-temporal overlap of reference and system objects
$E_O = TBD$; a constant to weight object detection congruence
$O_c(I_{R_i}, I_{S_j})$ = 1 - minMODE; the object detection congruence function between a reference and system output instance defined in Section 6.2.2.

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

6.2.2. Spatial Object Detection per Activity Instance

For the object detection, we employ the N_MODE (Normalized Multiple Object Detection Error) metrics described in [4][5]. N_MODE evaluates the relative number of false positives and missed detections for

objects per activity instance. Note that these metrics are applied to the frames in correspondence instance pairs only. The object detection is evaluated for the frames within the intersection of the reference and system output for the aligned instance pair and the metric is only based on the required objects (e.g., person, vehicle).

The metric includes the object alignment for the frame-level object mapping between reference bounding boxes and system output bounding boxes using the Hungarian algorithm (see the paper [4] for the detailed descriptions). For object alignment procedure, the following new kernel function K_O is used. See Section 6.1.1 for the previous variable definitions.

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

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

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

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

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

where,

$$OP_c(B_{s_j}) = \frac{OP(B_{s_j}) - OP_{min}(S_{OP})}{OP_{max}(S_{OP}) - OP_{min}(S_{OP})}$$

B_{R_i} : the i^{th} reference bounding box of the objects
B_{S_j} : the j^{th} system output bounding box of the objects
$Spatial_{IoU}(B_{R_i}, B_{S_j})$: $Intersection(B_{R_i}, B_{S_j})$ over $Union(B_{R_i}, B_{S_j})$ in a spatial domain
$\Delta Spatial_{IoU}$ = TBD; the fixed spatial IoU threshold
$E_{IoU} = TBD$; a constant to weight overlap ratio congruence
$E_{OP} = TBD$; a constant to weight object presence confidence score congruence
$OP_c(B_{S_j})$: a presence confidence score congruence of the system object B_{S_j}
$OP(B_{S_j})$: the presence confidence score of object B_{S_j}
S_{OP} : the system presence confidence score that indicates the confidence that the object has been identified
$OP_{min}(S_{OP})$: the minimum presence confidence score from a set of object confidence scores per instance, S_{OP}
$OP_{max}(S_{OP})$: the maximum presence confidence score from a set of object confidence scores per instance, S_{OP}

The confusion matrix for each frame t is calculated with presence confidence scores of object bounding boxes, referred to as the confidence threshold τ . Note that we only consider frames within the intersection of the reference and system temporal localizations.

- Correct Detection ($CD_t(\tau)$): the count of reference and system output object bounding boxes that are mapped to each other for frame t with bounding box confidence score >= τ .
- Missed Detection ($MD_t(\tau)$): the count of reference bounding boxes not mapped to a system output bounding box at bounding box confidence score >= τ .
- False Alarm ($FA_t(\tau)$): the count of system bounding boxes with bounding box confidence score >= τ not aligned to reference bounding box.

- Correct Rejection: this metric is not calculated in this evaluation.

The task requires systems to provide confidence scores of object bounding box, thus, N_MODE (Normalized Multiple Object Detection Error) is computed for all object bounding box thresholds τ to determine whether the system correctly detects the presence of objects over time per the corresponding instance pair and is defined as:

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

where τ is the object confidence score threshold. For the instance pair, the minimum MODE value (*minMODE)* and $P_{Miss}@Rate_{FA}$ point values are used as the object detection performance. $P_{Miss}@Rate_{FA}$ will also be reported at the activity level, and for all activities, by concatenating the pair-level object alignments. In this case, the normalization of your system's object presence confidence scores across activity instances will be reflected in your score . We will report P_{Miss} at fixed $Rate_{FA}$ points including: 0.5, 0.2, 0.1, and 0.033 false alarms. For Phase 1 (1.A and 1.B) evaluations, system performance for object detection is primarily evaluated at one operating point, P_{Miss} at RFA=0.5.

$MD_t(\tau)$: the object missed detections in frame t at the object confidence score threshold τ
$FA_t(\tau)$: the object false alarms in frame t at the object confidence score threshold τ
C_{MD} : the cost function for missed detections
C_{FA} : 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

6.2.3. Spatial Object Localization (Secondary Metrics)

This secondary task will not be evaluated in the phases 1.A and 1.B. Thus, the metrics will be defined for future evaluations to measure the spatial accuracy of object bounding box placements.

6.3. Activity Object Detection and Tracking Metrics (AODT)

For this task, the metrics below evaluate system performance on whether the system 1) correctly detects the presence of the target activity, 2) correctly detects the presence of the required objects in that activity (object type and bounding box), and 3) correctly tracks these objects over time.

6.3.1. Activity Object Detection and Tracking (Primary Metric)

For the AODT task, the primary metric stays the same as AD and AOD, $P_{Miss}@Rate_{FA}$, but the alignment step (see the alignment procedure above in Section 6.1.1) uses an extended kernel function that incorporates object tracking as a requirement for correctness (i.e., correspondence).

The AODT task uses the same bipartite graph matching algorithm for reference and system activity instance alignment but uses an object tracking congruence function, $OT_c(I_{R_i}, I_{S_j})$, in the kernel function K.

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

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

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

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

$$\varnothing \; if \; OT_c(I_{R_i}, I_{S_j}) < \Delta OT_c$$

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

See Section 6.1.1 for the previous variable definitions
$E_{OT} = TBD$; a constant to weight object type tracking congruence
$OT_c(I_{R_i}, I_{S_j})$ = minMOTE; the object tracking congruence function between a reference and system output instance. The metric calculation will be constrained to only system object tracks associated with the system activity instance and the object tracks for the reference activity instance.
ΔOT_c = TBD; constant setting the minimum tracking score of reference and system objects

6.3.2. Object Tracking per Activity Instance

For the object tracking metrics, we employ the MOTE (Multiple Objects Tracking Error) metric described in [4][5] which takes the identity of objects into account and produces a score based on the count of false alarms, missed detections, and identity switches. Note that these metrics are applied to the frames in corresponding instance pairs. The object tracking is evaluated for the frames within the intersection of the reference and system output for the aligned instance pair and the metric is only based on the required objects (e.g., person, vehicle).

The metric includes the object alignment described-above for the frame-level object mapping between reference bounding boxes and system output bounding boxes using the Hungarian algorithm (see Section 6.2.2 for the detailed descriptions). Similar to the spatial object detection in Section 6.2.2, the task requires systems to provide presence confidence scores of object detection bounding boxes. Thus, the confusion matrix and IDSwitches values for each frame t are calculated with presence confidence scores of object bounding boxes, referred to as the confidence threshold τ . MOTE (Multiple Object Tracking Error) is computed for all object bounding box thresholds τ to determine whether the system correctly detects and tracks multiple objects over time. The metric is defined as:

$$MOTE(\tau) = \frac{\sum_{t=1}^{N_{frames}} C_{MD} * MD_t(\tau) + C_{FA} * FA_t(\tau) + C_{ID} * IDSwitches_t(\tau)}{\sum_{t=1}^{N_{frames}} N_R^t}$$

where τ is the object confidence score threshold. For the aligned activity instance pair, the minimum MOTE value (*minMOTE)* is used as the tracking performance. In this evaluation, C_{MD}, C_{FA}, and C_{ID} are equal to 1.

$MD_t(\tau)$: the object missed detections in frame t at the object confidence score threshold τ
$FA_t(\tau)$: the object false alarms in frame t at the object confidence score threshold τ
C_{MD} : the cost parameter for missed detections
C_{FA} : the cost parameter for false alarms
C_{ID} : the cost parameter for object ID switches (tracking error)
$IDSwitches_t(\tau)$: the number of object ID mismatches in frame t considering the mapping in frame (t-1) at the object confidence score threshold τ
N_R^t : the number of reference objects in frame t
N_{frames} : the number of frames in the sequence for the instance

6.3.3. Spatio-Temporal Localization of Objects (Secondary Metric)

This secondary task will not be evaluated in the phases 1.A and 1.B. Thus, the metrics will be defined for future evaluations to measure the spatio-temporal accuracy of object bounding box placements.

6.5. System Information

6.5.1. System Description

A brief technical description of your system. Please see the detailed format in Appendix A-a System Descriptions.

6.5.2. 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.

6.5.2.1. Speed Measures and Requirements

For the ActEV 1.A and Leaderboard evaluation the participants will report the processing speed per video stream compared to real-time by running only on one node of their system for each task separately. For this program, real-time processing refers to processing at the same rate as the input video.

For the 1.B self-reported evaluation the open participants and performers will report the processing speed per video stream compared to real-time by running only on one node of their system for each task separately. The primary 1.B system is required and has to meet the time requirements and the secondary system is optional and not time limited.

For Phase 1.B Independent Evaluation, NIST will run the participants and performer's system on one of the nodes of the NIST Independent Evaluation Infrastructure costing less than $10K (Hardware specification for 1.B are provided in Appendix C and more details for speed measures and requirements are in Appendix A).

6.5.3. Training Data and Knowledge Sources

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

6.5.4. System References

List pertinent references, if any.

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 ActEV2018 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 multiples 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 allows 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 strings 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-a 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 4 Speed Measures and Requirements

For the 1.A evaluation and the Leaderboard evaluation the open participants and performers will report the processing speed per video stream compared to real-time by running only on one node of their system for each task separately. For this program, real-time processing refers to processing at the same rate as the input video.

For the 1.B self-reported evaluation the open participants and performers will report the processing speed per video stream compared to real-time by running only on one node of their system for each task separately. The primary 1.B system is required and has to meet the time requirements (for 19 target activities the system should not be more than 20 times slower than realtime) and the secondary system is optional and not time limited.

For Phase 1.B Independent Evaluation, NIST will run the participants and performers system on one of the nodes of the NIST Independent Evaluation Infrastructure costing less than $10K (Hardware specification are provided in Appendix C). The systems should not be slower than five times that of realtime for the five activities, then for 19 target activities, the system should not be more than 20 times slower than realtime. In the event that a system exceeds the speed requirements by 20%, NIST will not be in a position to complete evaluation of that system.

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-b 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-c 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/

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

JSON SCHEMA FOR SYSTEM OUTPUT FILE

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

Appendix C: Infrastructure (Hardware and Virtual Machine specification)

Scoring Server

The open participants and performers will submit their system output in the Json file format described earlier to an online web based evaluation server application at NIST. The PI for each performer team 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 scores will be manually reviewed by the DIVA T&E team prior to dissemination to the PI or performer team. There will be a dry run of the scoring server.

NIST Independent Evaluation Infrastructure Specification

Hardware specification:

- CPU - 16 cores
- Memory - 128GB
- GPU – 4 x NVIDIA® GeForce GTX 1080 Ti GPU - 11GB
- Root disk size - 40GB
- 250GB SSD cache
- Storage Volume- 1TB (variable)
- Supplied object store (read only) for source video
-

Independent Evaluation Infrastructure and Delivery of Software

The open participants and performers will deliver their algorithms that are compatible with the CLI protocol to NIST. The purpose is to test for compatibility and to ensure that the test results that the participants and performers obtained on the "validation dataset" when running on their own server match what NIST is getting when they run it on the Independent Evaluation Infrastructure.

Appendix D: Definitions of Activity and Required objects [6]

For the ActEV leaderboard evaluation, the definitions of the 19 target activity and the objects associated with the activity are described below [6]. The target objects for the ActEV evaluations (1.A, Leaderboard, and 1.B) are P = {Person} and V = {Construction_Vehicle, Vehicle}.

Closing

Closing Description: A person closing the door to a vehicle or facility.
Start: The event begins 1 s before the door starts to move.
End: The event ends after the door stops moving. People in cars who close the car door from within is a closing event if you can still see the person within the car. If the person is not visible once they are in the car, then the closing should not be annotated as an event.
Objects associated with the activity : Person; and Door or Vehicle

Closing_trunk

Close Trunk Description: A person closing a trunk. See Open Trunk (above) for definition of trunk and special cases.
Start: The event begins 1 s before the trunk starts to move.
End: The event ends after the trunk has stopped moving.
Objects associated with the activity: Person; and Vehicle

Entering

Entering Description: A person entering (going into or getting into) a vehicle or facility.
Start: The event begins 1 s before the door moves or if there is no door, the event begins 1 s before the person's body is inside the vehicle/facility.
End: The event ends when the person is in the vehicle/facility and the door (if present) is shut.
Notes: A facility is defined as a structure built, installed or established to serve a particular purpose. This facility must have an object track (e.g., door or doorway) for the person to enter through. The two necessary tracks included in this event are
(1) the person entering and (2) the vehicle or the object for entering a facility (e.g., door). A special case of "entering" is mounting a motorized vehicle (e.g., motorcycle, powered scooter).
Note 2 : No special activity for standing or crouching when entering or exiting a vehicle. Whenever the person starts standing or walking, annotate as usual, but once they stop lateral motion and start bending down to get into out of the car, they've stopped both standing and walking, so no activity. Sitting in car when entering or exiting is only if sitting is visible for >10 frames.
Objects associated with the activity: Person; and Door or Vehicle

Exiting

Exiting Description: A person exiting a vehicle or facility. See entering for definition of facility.
Start: The event begins 1 s before the door moves or if there is no door, the event begins 1 s before half of the person's body is outside the vehicle/facility.
End: The event ends 1 s after the person has exited the vehicle/facility.
Note: A special case of "exiting" is dismounting a motorized vehicle (e.g., motorcycle, motorized scooter).
Objects associated with the activity: Person; and Door or Vehicle

Loading

Loading Description: An object moving from person to vehicle.
Start: The event begins 2 s before the cargo to be loaded is extended toward the vehicle (i.e., before a person's posture changes from one of "carrying" to one of "loading").
End: The event ends after the cargo is placed into the vehicle and the person-cargo contact is lost. In the event of occlusion, it ends when the loss of contact is visible.
Note: The two necessary tracks included in this event are the person performing the (un)loading and the vehicle/cart being (un)loaded. Additionally, if the items being loaded are at least half the person's size or large enough to substantially modify the person's gait (as defined in the Carrying activity -- 4.7), then they should be individually tracked as Props and included in the event. "Fiddling" with the object being (un)loaded is still part of the (un)loading process.
Objects associated with the activity: Person; and Vehicle

Open_Trunk

Open Trunk Description: A person opening a trunk. A trunk is defined as a container designed to store non-human cargo on a vehicle.
Start: The event begins 1 s before the trunk starts to move.
End: The event ends after the trunk has stopped moving.
Notes: A trunk does not need to have a lid or open from above. So the back/bed of a truck is a trunk and dropping the tailgate is the equivalent of opening a trunk. Additionally, opening the double doors on the back of a van is the equivalent of opening a trunk.
Objects associated with the activity: Person; and Vehicle

Opening

Opening Description: A person opening the door to a vehicle or facility.
Start: The event begins 1 s before the door starts to move.
End: The event ends after the door stops moving.
Note: The two necessary tracks included in this event are (1) the person opening the door and (2) the vehicle or the object for a facility (e.g., door). The vehicle door does not need to be independently annotated because the vehicle itself is a track which can be coupled to the person in this event. This event often overlaps with entering/exiting; however, can be independent or absent from these events.
Note 2: Opening clarification: When opening a car door, the event ends when the when the door stops moving from being opened. This is distinguished from someone opening a car door, then leaning on the door when they exit and the door wiggles.
The wiggling is not part of opening, even though it is in fact moving.
Objects associated with the activity : Person; and Door or Vehicle

Transport_HeavyCarry

Transport Large Object or Heavy Carry Description: A person or multiple people carrying an oversized or heavy object. This is characterized by the object being large enough (over half the size of the person) or heavy enough (where the person's gait has been substantially modified) to require being tracked separately.
Start: This event begins 1 s before the person (or the first person for multiple people) establishes contact with the object.
End: This event ends 1 s after the person (or the final person for multiple people) loses contact with the object.
Objects required : Person; and Prop

Unloading

Unloading Description: An object moving from vehicle to person.
Start: The event begins 2 s before the cargo begins to move. If the start of the event is occluded, then it begins when the cargo movement is first visible.
End: The event ends after the cargo is released. If the person holding the cargo begins to walk away from the vehicle, the event ends after 1 s of walking. If the door is closed on the vehicle, the event ends when the door is closed. If both of these things happen, the event ends at the earlier of the two events.
Note: See Loading above.
Objects associated with the activity: Person; and Vehicle

Vehicle_turning_left

Turning Left Description: A vehicle turning left or right is determined from the POV of the driver of the vehicle. The vehicle may not stop for more than 10 s during the turn.
Start: Annotation begins 1 s before vehicle has noticeably changed direction.
End: Annotation ends 1 s after the vehicle is no longer changing direction and linear motion has resumed.
Note: This event is determined after a reasonable interpretation of the video.
Objects associated with the activity : Vehicle

Vehicle_turning_right

Turning Right Description: A vehicle turning left or right is determined from the POV of the driver of the vehicle. The vehicle may not stop for more than 10 s during the turn.
Start: Annotation begins 1 s before vehicle has noticeably changed direction.
End: Annotation ends 1 s after the vehicle is no longer changing direction and linear motion has resumed.
Note: This event is determined after a reasonable interpretation of the video.
Objects associated with the activity : Vehicle

Vehicle_u_turn

U-Turn Description: A vehicle making a u-turn is defined as a turn of 180 and should give the appearance of a "U". A u-turn can be continuous or comprised of discrete events (e.g., a 3-point turn).The vehicle may not stop for more than 10 s during the u-turn.
Start: Annotation begins when the vehicle has ceased linear motion.
End: Annotation ends 1 s after the car has completed u-turn.
Note: This event is determined after a reasonable interpretation of the video. U-turns do not contain left and right turns (or start/stop in the case of K turns). U-turns are also annotated when going around something, like a bank of trees/shrubs.
Objects associated with the activity: Vehicle

Interacts

Interacts with Object Description: A person performs one of a wide variety of interactions with an object other than a vehicle or person that is not otherwise defined in this document.
Examples: Getting money from ATM, paying parking meter, mounting or dismounting a bike.
Start: This event begins 1 s before interaction.
End: This event ends 1 s after conclusion of interaction.
Note: This event usually begins/ends as soon as interaction between two tracks begins/ends, especially for

discrete interactions like touching a screen or kicking a post. Some interacts events are more fluid and should start/end 1s before/after the interaction, currently the only exception is mounting/dismounting a bike. Must be clear an interaction is occurring.
Objects associated with the activity: Person; and Object of any type other than person or vehicle

Pull

Pull Description: A person exerting a force to cause motion toward. The two necessary tracks included in this event are the person pulling and object being pulled (Push/Pulled Object - See Active Object Type 3.5).
Start: As soon as the object is visibly moving or track begins if object already in motion.
End: As soon as the object is no longer moving or the person loses contact with the object being pulled. In the event of occlusion, the event will end when the loss of contact is visible.
Objects required : Person; and Push/Pulled Object

Riding

Riding Description: A person riding a "bike" (i.e., any one of the variety of human powered vehicles where the person is still visible but their movement is modified).
Note: The two necessary tracks included in this event are (1) the person and (2) the "bike" they are riding. Events for Riding, Pushing and Pulling are used to couple the person and "bike" tracks.
Start: This event begins when the person's motion is modified by the "bike", or upon entering the FOV if the person is already riding the "bike".
End: This event ends when the person's motion is no longer modified by the "bike", or upon exiting the FOV
Objects associated with the activity: Person(s);

Talking

Talking Description: A person talking to another person in a face-to-face arrangement between n + 1 people.
Start: This event begins when the face-to-face arrangement is initiated.
End: This event ends when the face-to-face arrangement is broken.
Objects associated with the activity: Person(s);

Activity_carrying

Carrying Description: A person carrying an object up to half the size of the person, where the person's gait has not been substantially modified. The object may be carried in either hand, with both hands, or on one's back.
Examples: Carrying a Backpack, Purse, Briefcase, or Box.
Counter-examples: "Incidental carrying" such as a sheet of paper or a file folder such that the person's arm motion is not affected by the payload.
Start: Annotation begins in one of two ways: (1) when the person who will be carrying the object makes contact with the object, or (2) when the track begins, if the person is already carrying the object (e.g., backpack or purse).
End: Annotation ends when contact with the object is broken.
Note: If a carried object (e.g., purse, backpack, box) is separated from the individual, a new track for that object ("Prop") will be created. The events, pickup, drop, and set down will be used to couple/decouple the person and object.
Objects associated with the activity: Person(s);

Specialized_talking_phone

Talking On Phone Description: A person talking on a cell phone where the phone is being held on the side of the head. This activity should apply to the motion of putting one's hand up to the side of their head regardless of the presence of a phone in hand.
Start: Annotation should begin when hand makes motion toward side of head.
End: Annotation should end 1 s after hand leaves side of head.
Objects associated with the activity: Person(s);

Specialized_texting_phone

Texting On Phone Description: A person texting on a cell phone. This applies to any situation when the phone is in front of the person's face (as opposed to along the side of the head) and they are using it. This includes using the phone with thumbs and fingers or video chatting.
Start: Annotation should begin 1 s before "texting" is observed.
End: Annotation should end 1 s after last instance of "texting" is observed.
Objects associated with the activity: Person(s);

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.

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.