NIST AI Technology Evaluation Test Specification
Public Safety Visual Event Recognition

Version 1.0— May 28, 2026

Test Version:	1.0
Task:	Detection
Input:	Multiple Images and Text
Output:	Text
Task Champion(s):	Jim Golden
Dataset:	Gumby v1.0
Metric:	A weighted detection cost function will be computed.

Introduction

When public safety events occur, a swift and effective response is paramount, and failure to respond quickly and appropriately can result in major disruptions, as well as loss of lives and property that could otherwise be avoided. Real-time public safety imagery enables emergency personnel to optimally assess and respond to public safety events. However, the contiuous monitoring, especially a large number, of imagery sources is a known challenge for humans. The task described in this test represents a basic ability to identify public safety events from imagery data. As models demonstrate their abilities and improve over time, increasingly complex tasks in support of public safety event recognition, potentially including additional modalities, will be added to the test.

Task Definition

Given a set of three video key frames, automatically the model must output a yes or no decision whether the key frames provide evidence of a public safety event occurring.

Dataset Description

The test data consist of key frames selected from videos.

1. *Modalities:* Images
2. *Data creation/collection date(s):* April 2026
3. *Data Size:* $\sim$ 9 GB (9000 images in sets of three per potential event)
4. *Reference labels:* Human annotators watched and annotated the (scripted) videos
5. *Other pertinent features:* Source dataset is video, and mostly filmed at distance

Testing Protocol

Provide sets of three key frames in chronological order, where, if there is evidence of an event, the first frame corresponds to the start, the second the middle, and the third the end of the event. Ask the model whether there is evidence of a public safety event occurring based on the key frames.

1. *Input:* Three images and a fixed prompt
2. *Output:* Decision (yes or no)
3. *Scoring procedure(s):* Get output from model, compare against ground truth, compute error rates and metric value

Metric and Score Reporting

The metric use will be a detection cost function. Each time the model detects evidence of a public saftety event when none is taking place (a "false alarm"), there is a cost associated with the dispatcher having to review the images despite no event taking place. Likewise, each time the model fails to detect evidence of a public saftety event when one is taking place (a "miss"), there is a cost associated with the dispatcher not being alerted to the event. The miss to false_alarm relative costs in this test are set to be 1 to 20. However, there are many more instances of there being no public safety event taking place, and thus the prior probability of public_safety_event in this test is set to be 1%. Thus the raw cost for the system will be computed by:

$$\begin{aligned} C_{Det} = & C_{Miss} \times P_{Event} \times P_{Miss} + \\ & C_{FalseAlarm} \times (1 - P_{Event}) \times P_{FalseAlarm}, \end{aligned} \tag{1}$$

where the function parameters can be found in Table 1.

C_{Miss}	$C_{FalseAlarm}$	P_{Event}
1	20	0.01

Table 1: AITE Public Safety Video Event Recognition Cost Parameters

To improve the interpretability of the cost function, it will be normalized by $C_{Default}$ which is defined as the best cost that could be obtained without processing the input data (i.e., by either always accepting or always rejecting the segment individual(s) as matching the target individual, whichever gives the lower cost), as follows

$$C_{Norm} = \frac{C_{Det}}{C_{Default}}, \tag{2}$$

where $C_{Default}$ is defined as

$$C_{Default} = min \begin{cases} C_{Miss} \times P_{Target}, \\ C_{FalseAlarm} \times (1 - P_{Target}). \end{cases} \tag{3}$$

Substituting the set of parameter values from Table 1 into (3) yields

$$C_{Default} = C_{Miss} \times P_{Target}. \tag{4}$$

Substituting C_{Det} and $C_{Default}$ in (2) with (1) and (4), respectively, along with some algebraic manipulations yields the final metric, C_{Norm_β}, defined as follows:

$$C_{Norm_\beta} = P_{Miss} + \beta \times P_{FalseAlarm}, \tag{5}$$

where β is defined as

$$\beta = \frac{C_{FalseAlarm}}{C_{Miss}} \times \frac{1 - P_{Target}}{P_{Target}}. \tag{6}$$

Uncertainty will be computed using bootstraping methods[1].

Change Logs

- Version 1; May 28, 2026; Initial Release

[1] Efron, Bradley. "Bootstrap methods: another look at the jackknife." Breakthroughs in statistics: Methodology and distribution. New York, NY: Springer New York, 1992. 569-593.