Media Forensics Challenge 2019 Evaluation Plan

Date: 2018-12-05

NIST MediFor Team

NIST

Table of Contents

LIST OF FIGURES

List of Tables

1 INTRODUCTION

This document describes the system evaluation tasks supported by the 2019 Media Forensics Challenge sponsored as part of the Defense Advanced Research Projects Agency (DARPA) Media Forensics (MediFor) program (http://www.darpa.mil/program/media-forensics). The Media Forensics Challenge 2019 (MFC2019) evaluation plan covers resources, task definitions, task conditions, file formats for system inputs and outputs, evaluation metrics, scoring procedures, and protocols for submitting results.

The Media Forensics Challenge is a media forensics evaluation to measure how well systems can automatically detect and locate manipulations in imagery (i.e., images and videos) as well as construct a phylogeny graph for a manipulated image using a pool of imagery.

Any questions or comments concerning the MFC2019 should be sent to mfc_poc@nist.gov.

2 TASKS AND CONDITIONS

In the MFC2019 evaluation, there are four tasks for systems that detect manipulated images and videos: manipulation detection and localization, splice detection and localization, provenance filtering, and provenance graph building. The tasks will be evaluated under two different conditions: image content only and image content plus metadata. For each task, the system will be prompted with a probe, an image or video that is the subject of the task question posed to the system.

2.1 TASKS

2.1.1 MANIPULATION DETECTION AND LOCALIZATION

For the manipulation detection and localization (MDL) task, the objective is to detect if a probe has been manipulated and, if so, to spatially localize the edits. Localization is encouraged but not required for MFC2019. Manipulation can be of many forms, including resizing, splicing, cloning, cropping, histogram equalization, etc.

For each trial, which consists of a single probe image or video, the MDL system must render a confidence score[1] with higher numbers indicating the probe image is more likely to have been manipulated. The primary metric for measuring detection performance will be Area Under the Receiver Operating Characteristic (ROC) Curve (AUC) (see Section 6.1.2) as well as Correct Detection (CD) at a False Alarm Rate (FAR) of 5% (see Section 6.1.3); additional metrics may be used.

2.1.1.1 IMAGE LOCALIZATION

For image localization, the system-rendered mask image for each trial must be relative to the probe image and must indicate the region(s) and confidence that the probe image was manipulated. The form of the system-provided masks is defined in Section 5.1.2.1. If the mask image for a trial is detected by a system to find no localizable content change, it can be omitted and is assumed to be empty. The reference mask for each true manipulation with localized content change is a layered reference mask[2], a lossless JPEG 2000[3] image in which, on each layer, a white pixel indicates the region has not been manipulated and a black pixel indicates the region

[1] The confidence score can be of any real number in the range $[0, 1]$. The confidence scores must be orderable across trials, but not systems.
[2] Defined in Section 3.5.2.
[3] https://jpeg.org/jpeg2000/index.html

has been manipulated. The reference mask for each true manipulation without localized content change is a reference mask, a JPEG 2000 image in which each pixel is white; that is, if there is no localized content change in a true manipulation, the mask is completely white. Each bit plane indicates a separate manipulation. Not all manipulations require localization output. Global operations affecting the entire image are not required for localization output because then the entire image is marked as manipulated; for example, a clone operation does require localization output while global histogram normalization does not. In the future, global operations may be addressed as a separate task. The primary metric for measuring image manipulation localization performance will be the Optimum Matthews Correlation Coefficient (MCC) (see Section 6.2.3); additional metrics may be used.

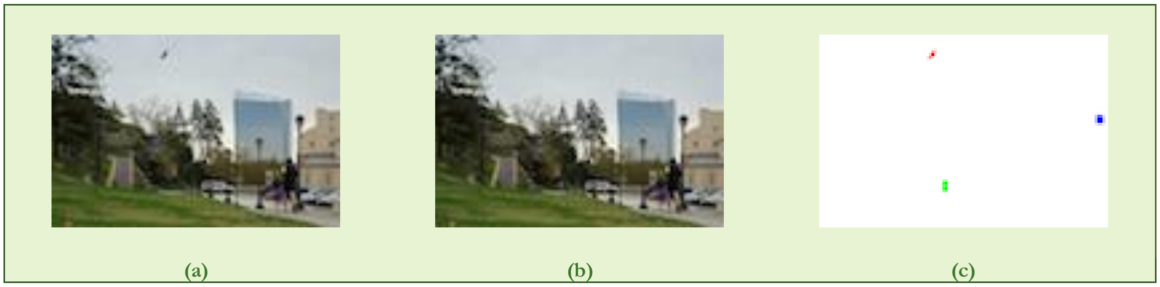

Figure 1: An example of a trial for the image manipulation detection task[4]

Figure 1shows an example of a manipulation detection trial. In this trial image (b) is the original image. Image (a) is created by removing a jogger, cloning a window, and splicing a hawk into the image. Each manipulation in the trial is indicated by a different color in the reference mask as shown in image (c). The removal of the jogger is indicated by the green color, the cloning of the window is indicated by the blue color, and the splicing in of the hawk is indicated by the red color.

2.1.1.2 Video Localization

Two types of video localization are temporal and spatial. For both types, the primary metric for measuring video manipulation localization performance will be the Optimum Matthews Correlation Coefficient (MCC) (see Section 6.2.3); additional metrics may be used.

For video temporal localization, relative to the probe video, the system must indicate the frame intervals and/or audio time segments that the probe video was manipulated, video or audio. If the video and/or audio temporal localization is determined by a system to find no content change, the set of intervals can be omitted and is assumed to be empty.

For video spatial localization, the system-rendered mask must be relative to the probe video and must indicate the region(s) of the probe that were manipulated. The form of the system-rendered masks is defined in Section 5.1.2.2. If the mask for a video trial is detected by a system to find no localizable visual content change, it can be omitted and is assumed to be empty. The reference mask[5] for each true manipulation with localized visual content change is a set of Hierarchical Data Format 5[6] (HDF5) files which contain disjoint intervals of

[4] In Figure 1, image (a) is a derivative of image (b) [4229350757_4f8bae3870_o.jpg (http://farm3.staticflickr.com/2694/4229350757_4f8bae3870_o.jpg) by michaelwm25] and of 5559691732_7d70e4b268_o.jpg (http://farm6.staticflickr.com/5306/5559691732_7d70e4b268_o.jpg) by BobMacInnes. All images are used under CC-BY 2.0 (https://creativecommons.org/licenses/by/2.0/).

[5] Defined in Section 3.5.2.

[6] https://www.hdfgroup.org/

frames where a white pixel indicates the region of the frame has not been manipulated and a black pixel indicates the region of the frame has been manipulated. If the spatial mask for a trial is detected by a system to find no localizable content change, it can be omitted and is assumed to be empty. Not all manipulations require localization output. Global operations affecting the entire visual content of the video are not required for spatial localization output because then the entire video is marked as manipulated; for example, a clone operation does require spatial localization output while global histogram normalization does not. In the future, global operations may be addressed as a separate task. Auditory operations are not required for spatial localization but are required for temporal localization.

2.1.1.3 Manipulation Mask Scoring Variations

MDL performance will be assessed using two methods: across all manipulation types present in the test collection, the default, and selectively focusing on manipulations of interest (i.e. selective scoring). Selective scoring is supported for image and video temporal localization in MFC2019. Video spatial localization is a pilot program in MFC2019, so selective scoring is not supported for the MFC2019 evaluation.

The selective manipulation type scoring protocol uses query to divide manipulations into two groups, selected manipulation types and un-selected manipulation types. Evaluated probes can be one of the following:

- contain only the selected manipulation type – the probe is scored as usual for both detection and localization.
- contain a mix of selected and un-selected manipulation types – the probe is scored as usual for detection. For localization, mask regions containing un-selected manipulation types are treated as no-score regions after a dilation with a kernel of 11 pixels.
- contain only un-selected manipulation types – the probe is not scored for both detection and localization.

Table 1 lays out the treatment of probes for both scoring protocols.

Table 1: Probe Treatment under Scoring Protocols

Variations of manipulations within probes	All Manipulation Scoring		Selective Manipulation Scoring	
	Detection Reference	Localization Reference	Detection Reference	Localization Reference
Only Selected	Target	FullMask	Target	FullMask
Selected and Un-selected	Target	FullMask	Target	FullMask(Sel) – NoScore(UnSel)
Only Un-selected	Target	FullMask	NotScored	NotScored
Non - Manipulated	NonTarget	NotScored	NonTarget	NotScored

Under the Detection Reference columns, “Target” refers to manipulated probes that are scored. “NonTarget” refers to non-manipulated probes that are scored. “NotScored” refers to manipulated probes that do not contain the selected manipulation type(s) and are therefore not scored.

Under the Localization Reference columns, "FullMask" indicates that the entire mask of the probe is scored. "NotScored" indicates that the probe is non-manipulated or that the probe is manipulated but contains no selected manipulation type(s) and thus is not scored at all. "NoScore" indicates that the probe contains manipulation type(s) that were not selected as well as manipulation type(s) that were selected; therefore, the manipulation type(s) that were not selected are not scored when scoring the rest of the mask.

2.1.2 Splice Detection and Localization

For the splice detection and localization (SDL) task, the objective is to detect if a region of a given potential donor image has been spliced into a probe image and, if so, provide the mask images indicating the region(s) of the donor image that were spliced into the probe and the region(s) of the probe image that were spliced from the donor.

For each splice detection trial consisting of a pair of probe and donor images, the system must render a confidence score indicating how likely it is that a region from the donor image has been spliced into the probe image. The primary metric for measuring detection performance will be AUC (see Section 6.1.2) as well as CD at FAR of 5% (see Section 6.1.3); additional metrics may be used. Probes will include imagery containing a splice, imagery containing other manipulations, and non-manipulated imagery.

For localization, the system must also render two masks: one indicating the region(s) of the donor that was copied and one indicating the region(s) of the probe that was pasted from the donor. The form of the system-provided masks is defined in Section 5.1.2.1. If either mask is detected by a system to find no localizable content change, they can be omitted and are assumed to be empty. The reference mask for the probe image of a true trial is a manipulation colorized reference mask restricted to the spliced content. The reference mask for the donor image of a true trial is a colorized mask restricted to the region(s) spliced into the probe image. The primary metric for measuring the performance will be the Optimum MCC (see Section 6.2.3) and tabulated separately for probe and donor images; additional metrics may be used.

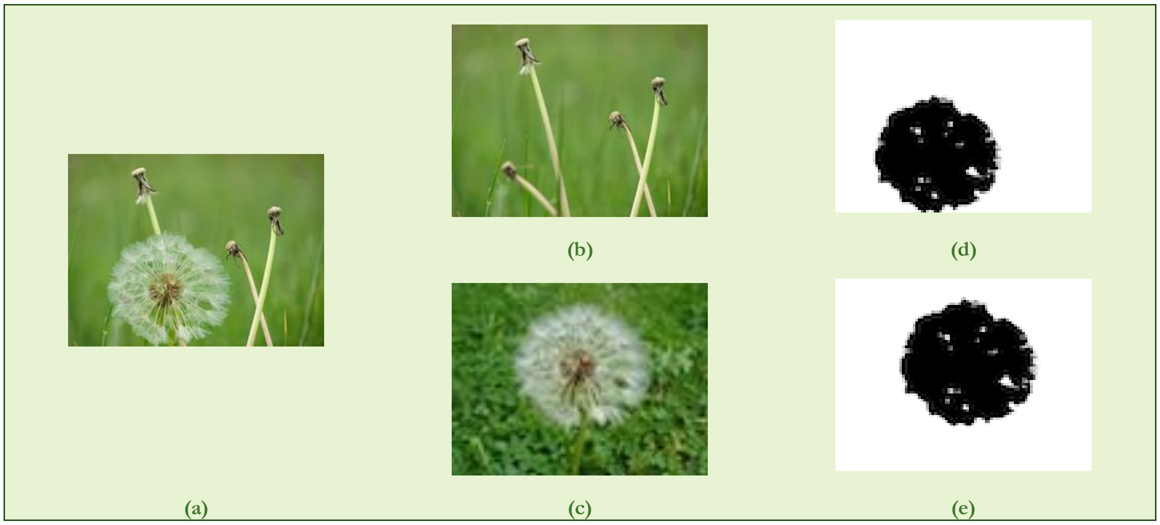

Figure 2: An example of a trial for the splice detection task[7]

[7] In Figure 2, image (a) is a derivative of image (b) [2500891663_010a955676_o.jpg (http://farm4.staticflickr.com/3014/2500891663_010a955676_o.jpg) by bortescristian] and image (c) [5738034619_f06b4b3964_o.jpg (http://farm4.staticflickr.com/3504/5738034619_f06b4b3964_o.jpg) by Violette79]. All images are used under CC-BY 2.0 (https://creativecommons.org/licenses/by/2.0/).

Figure 2 shows an example of a splice detection trial. In this journal, image (b) is the original image. Image(a) is created by splicing the dandelion from image (c) into image (b). In this trial, image(a) is the probe and image (c) is the potential donor. The system must render a confidence score indicating the strength of evidence a region of the potential donor was spliced into the probe as well as two masks: one identifying the region(s) of the potential donor spliced into the probe (image (e)) and another identifying the region(s) of the probe spliced from the potential donor (image (d)).

2.1.3 Provenance Filtering

For the provenance filtering task, the objective is to return up to n images from the provided world data set that includes all images (its ancestors and descendants) present from a probe image's genealogy graph; note that some images from the probe's whole genealogy graph may be missing from the world data set. A confidence score indicating how likely the image is in the genealogy graph must be provided for each of the returned images. Manipulation(s) can be of any form, including resizing, splicing, cloning, etc. In MFC2019, performance will primarily be measured by recall at $n = 300$ over all probes. Additional metrics may be used. Metrics will be evaluated under the full graph condition (see Section 2.1.4.4).

No masks are necessary or collected for this task in MFC2019.

2.1.4 Provenance Graph Building

For the provenance graph building task (or provenance task) the system must construct and label a provenance (phylogeny) graph for a probe image by finding the ancestor and descendent images present within the world data set; note that some images from the probe's whole genealogy graph may be missing from the world data set. Probes can be: a final manipulated image (of any form), a base image or intermediate modified image (images with modified descendants in the world data set), donor image (images whose content is inserted into a modified image in the world data set), or non-modified images.

No mask images are necessary or collected for this task in MFC2019.

The form of the phylogeny graph is described in Appendix C. The primary metric for measuring provenance graph building will be $\mathrm{sim}_{\mathrm{NLO}}$ (see Section 7). Metrics will be evaluated under two conditions: direct graph and full graph.

Depending on the pool of world images, there are three variations of this task: end-to-end, oracle filter (with distractors), and oracle filter (without distractors).

2.1.4.1 End-to-End Provenance Graph Building

For the end-to-end provenance graph building task, the objective is to produce a provenance graph for a probe image using the five million-image world collection as input.

2.1.4.2 Oracle Filter (With Distractors) Provenance Graph Building

For the oracle filter (with distractors) provenance graph building task, the objective is to produce a provenance graph for a probe image using a NIST-provided small collection of 300 images, which includes distractors, as input.

2.1.4.3 Oracle Filter (Without Distractors) Provenance Graph Building

For the oracle filter (without distractors) provenance graph building task, the objective is to produce a provenance graph for a probe image using a NIST-provided collection of the related imagery (with no distractors) as input.

2.1.4.4 Graph Building Evaluation Conditions

For both variations of the provenance graph building task, performance will be measured under two conditions: direct graph or full graph. Under the direct graph condition, the node set (the set of images in the provenance graph) is restricted to ancestors and descendants of the probe, i.e., nodes in which there is a directed path to or from the probe image. Under the full graph condition, there is no restriction on the node set, and all images related to the probe in the world data, i.e., there exists an undirected path between the image and the probe in the provenance graph, set must be in the graph. Systems are expected to return the entire provenance graph.

2.1.5 Event Verification

For the event verification task, the objective is to determine if a probe is from a claimed event, given a collection of images and videos from the event. For each event verification trial consisting of a probe and an event, the systems must render a confidence score indicating how likely it is that the probe is from the event. The primary metric for measuring detection performance will be AUC (see Section 6.1.2) as well as CD at FAR of 5% (see Section 6.1.3); additional metrics may be used.

For all conditions (Section 2.3) for this task, systems are not allowed to use camera type or serial numbers.

2.1.6 Camera Verification

For the camera verification task, the objective is to determine if a probe is from a claimed camera fingerprint given a pool of the camera device IDs. For each camera verification trial consisting of a probe and an alleged high provenance (HP) camera device ID, the systems must render a confidence score indicating how likely it is that the probe was captured with the camera. The probe may be an image or video, and the pool for a camera device ID may contain images or videos. Depending on the composition of the pool of the camera device ID (training set) and the set of probes (testing set), there are five training-testing conditions; when results are submitted, the training-testing condition must be indicated.

- Training Sets
  - Image: only images are in the training set
  - Video: only videos are in the training set
  - Multimedia: both images and videos are in the training set
- Testing Sets
  - Image: only images are in the testing set
  - Video: only videos are in the testing set

The primary metric for measuring detection performance will be AUC (see Section 6.1.2) as well as CD at FAR of 5% (see Section 6.1.3); additional metrics may be used.

For localization, the system-rendered mask image for each trial must be relative to the probe-camera pair and must indicate the region(s) and confidence of the probe image captured by the camera. The form of the system-provided masks is defined in Section 5.1.2.1. If the mask image for a trial is detected by a system to find no localizable content change, it can be omitted and is assumed to be empty. The reference mask for each true manipulation with localized content change is a layered reference mask[8], a JPEG 2000 image in which, on each layer, a white pixel indicates the region has not been manipulated and a black pixel indicates the region has been manipulated. The reference mask for each true manipulation without localized content change is a reference mask, a JPEG 2000 image in which each pixel is white; that is, if there is no localized content change in a true manipulation, the mask is completely white. Each bit plane indicates a separate manipulation. Not all manipulations require localization output. Global operations affecting the entire image are not required for localization output because the entire image is marked as manipulated; for example, a clone operation does require localization output while global histogram normalization does not. In the future, global operations may be addressed as a separate task. The primary metric for measuring image manipulation localization performance will be the Optimum Matthews Correlation Coefficient (MCC) (see Section 6.2.3); additional metrics may be used.

2.2 OptOut Evaluation Protocol

For each trial in each task, the system may opt out of the trial. The system may also opt out of only one portion of the task for the trial, i.e. detection or localization. If a system utilizes the OptOut system response (see status fields in Section 5.1.1.3), NIST will report the system's Trial Response Rate (TRR; the fraction of trials for which a response was provided), performance measures over all trials (regardless of the trial's opt out status to set the context of system performance against the whole data set), and performance measures over the processed trials. The confidence score for non-processed or opted out trials must be 0.

For each system-rendered mask, the system may opt out of specific areas of a mask by designating a pixel-value (see ProbeOptOutPixelValue and DonorOptOutPixelValue fields in Section 5.1.1.3). Pixels with this probe-specific value will not be scored.

The process for determining which trials and pixels to opt out must be documented in the system description (Appendix A-a).

2.3 Conditions

2.3.1 Image Only

For the image only condition, ConditionID: **ImgOnly**, the system is only allowed to use the pixel-based content for images as input. No image header or other information must be used.

2.3.2 Image and Metadata

For the image and metadata condition, ConditionID: **ImgMeta**, the system is allowed to use metadata, including image header or other information, in addition to the pixel-based content for the image, as input.

2.3.3 Video Only

For the video only condition, ConditionID: **VidOnly**, the system is only allowed to use the pixel-based content for videos and audio if it exists as input. No video header or other information must be used.

[8] Defined in Section 3.5.2.

2.3.4 Video and Metadata

For the video and metadata condition, ConditionID: **VidMeta**, the system is allowed to use metadata, including video header or other information, in addition to the pixel-based content for the video and audio if it exists, as input.

2.4 Protocol

All trials, i.e., probes, must be processed independently of each other within a given task and across all tasks, meaning content extracted from probe data must not affect another probe.

Systems may pre-index the world data set for the provenance filtering and provenance graph building tasks and reuse the index so long as the index is static before probe processing.

Systems may pre-index the donor images for the splice detection and localization task so long as the index is static before probes are processed.

MFC evaluations do not provide training sets for MDL, SDL, Provenance Filtering, or Provenance Graph Building tasks. Previously released development data and evaluation data is not designed for training purposes for future evaluation cycles; performers use this data for training sets at their own risk.

All machine learning or statistical analysis algorithms must complete training, model selection, and tuning prior to running the MFC2019 test data.

3 Data Resources

Each MFC2019 data set consists of up to five main directories: ‘probe’, ‘world’, ‘documents’, ‘indexes’, and ‘reference’. They are explained below.

3.1 Probe Directory

The MFC2019 ‘probe’ directory contains images and videos that will be forensically analyzed. The images and videos may be either manipulated or non-manipulated. In MFC2019, images and videos may be of any format. For the MDL task, there are 50 000 images and 5 000 videos. For the SDL task, there are 50 000 images. For the provenance tasks (filtering and graph building), there are 50 000 images.

3.2 World Directory

The MFC2019 ‘world’ directory contains images and videos to simulate a real-world collection of media of unknown provenance. The directory may contain images and videos used as donors for some of the probes. In MFC2019, images and videos may be of any format. There are 5 000 000 images in the world directory.

3.3 Documents Directory

The MFC2019 ‘documents’ directory contains additional documentation provided with the data set.

3.4 Indexes Directory

The MFC2019 'indexes' directory contains a system index file for each task. An index file is a comma-separated value (CSV) file which lists the images or videos a system must process (see Section 4.1 and Appendix B for details).

3.5 Reference Directory

The MFC2019 'reference' directory contains a subdirectory for each evaluation task, i.e. manipulation detection, splice detection, provenance filtering, provenance, event verification, or camera verification. Within each detection directory are two types of data: (1) the reference files that contain the "ground-truth" and metadata about trial probes and (2) a subdirectory containing the reference masks. Within the provenance filtering directory is one file: the reference file that contains the ground-truth.

3.5.1 Reference Files for Detection Tasks

Three files constitute the reference files for the detection tasks. The main reference file, following the naming convention <DataSet_Name>-<TaskID>-ref.csv, contains seven columns that describe each trial. Additional columns, documented in the data release, will be used for analysis.

TaskID	The type of system output, e.g. "manipulation"
ProbeFileID	The ID of the probe, e.g., MFC2019_6209
ProbeFileName	The partial path name to the probe file (relative to the top node of the data distribution), e.g. probe/MFC2019_9369.jpg
IsTarget	Boolean indicating if the probe is a manipulated image, i.e. "Y" | "N"
ProbeMaskFileName	The partial path name to the manipulation mask for the probe (relative to the top node of the data distribution), e.g. reference/splice/mask/MFC2019_8774.png if IsTarget = "Y", blank otherwise (i.e., no content).
BaseFileName	The partial path name to the base image within the world data set (relative to the top node of the data distribution), e.g. world/MFC2019_8806.tif if IsTarget = "Y", blank otherwise.
JournalName	The name of the manipulation journal for which the probe was extracted, e.g. oof7oxgiqjprd4ou4lq75wtnvdlmwhkk if IsTarget = "Y", blank otherwise.

For the video MDL task, there are two additional columns:

FrameCount	The total number of frames in the probe, e.g. 1200
FrameRate	The number of frames displayed per second in the probe, e.g. 24

For the SDL task, there are 3 additional columns:

DonorFileID	The ID of the donor image
DonorFileName	The partial path name to the donor image (relative to the top node of the data distribution), e.g. world/MFC2019_492_3.png if IsTarget = "Y", blank otherwise.
DonorMaskFileName	The partial path name to the donor mask (relative to the top node of the data distribution), e.g. reference/splice/mask/MFC2019_492_mask.png if IsTarget = "Y", blank otherwise.

For each probe for a given detection TaskID, the file <DataSet_Name>-<TaskID>-ref-probejournaljoin.csv documents the journal from which the probe came as well as the operation(s), identified by the before-operation-node and after-operation-node, referenced in the <DataSet_Name>-<TaskID>-ref-journalmask.csv. Journals may include sub-graphs that do not apply to a given probe; only entries that pertain to a given probe are in the probejournaljoin file.

ProbeFileID	Same as above
JournalName	Same as above
StartNodeID	The starting NodeID within the journal whose operation is included in the probe, e.g. if77i8v5clk3g2btmpz038hhrnx499s3-TGT-01
EndNodeID	The starting NodeID within the journal whose operation is included in the probe, e.g. if77i8v5clk3g2btmpz038hhrnx499s3-TGT-02-FILL
BitPlane	The bit position of the manipulation in the JPEG2000 mask, e.g. $BP = 2$; the (i, j)-th pixel was modified by the operation if $1 = pixel_{i,j} \wedge 2^{BP-1}$

For the video MDL task, there is one additional column:

VideoTaskDesignation	Defines which video task can be supported by the probe, i.e. "spatial", "spatial-temporal", or "temporal"

The file <DataSet_Name>-<TaskID>-ref-journalmask.csv documents all masks for each operation in the journal, including manipulations not necessarily included in the probe. Each row is an operation; for localizable operations, a color is provided.

JournalName	Same as above
StartNodeID	Same as above
EndNodeID	Same as above
Operation	The manipulation operation type from the journal JSON file, e.g. "PasteSplice"
Color	The RGB color as a triplet of integers between 0 and 255, e.g. 255 10 0
Purpose	The semantic purpose of the manipulation, e.g. an object "remove" can be accomplished with several types of operations
OperationArgument	Arguments supplied with the given operation, e.g. "natural object"
AutoDesignName	The design operation type applied to generate output of auto manipulations, e.g. LaunderingFacebook_Option_1
PlugInName	The name of the specific plug-in used for auto manipulations, e.g. "GaussianLocalBlur"
PlugInArguments	The arguments used for the plug-in, e.g. "kernelsize 9"
SubCategory	Additional information in regards to the operation, e.g. GAN

For videos, there are 5 additional columns.

VideoTime	The time segments, given as millisecond intervals, where video manipulation occurs (each video starts at 0 ms), e.g. $[[0, 600], [1000, 7000], [1100, 1200]]$
VideoFrame	The frame segments, given as integer intervals, where video manipulation occurs (each video starts at Frame 1), e.g. $[[1, 20], [40, 230], [37, 40]]$
AudioTime	The time segments, given as millisecond intervals, where audio manipulation occurs (start at 0 ms), e.g. $[[300, 1200]]$

AudioSample	The sample segments, given as integer intervals, where audio manipulation occurs, e.g. $[[10, 40]]$
FrameTimeAdjustment	The number of milliseconds to add to the video time segments to adjust time alignment with frames, e.g. 3

3.5.2 Reference Mask

A reference mask is an image or series of frames used to represent which regions of an image or video have been manipulated.

For the image manipulation localization task, the mask is a JPEG 2000 image. A white pixel indicates that the region is not manipulated while a black pixel indicates that the region is manipulated in some way. Each bit plane indicates a separate manipulation. The reference mask can be filtered according to the types of manipulation a system detects.

For the video manipulation spatial localization task, the mask is an HDF5 file containing an interval of manipulated frames, disjoint from other manipulated intervals. A white pixel indicates that the region is not manipulated while a black pixel indicates that the region is manipulated in some way. Each bit plane indicates a separate manipulation. Video spatial localization is a pilot program in MFC2019, so selective scoring, filtering according to the types of manipulation a system detects, is not supported for the MFC2019 evaluation.

3.5.3 Reference Files for Provenance Tasks

Two files constitute the references files for the provenance tasks. The main reference file, following the naming convention <DataSet_Name>-<TaskID>-ref.csv, contains eight basic columns, as seen below that describe each trial. Additional columns, documented in the data release, will be used for analysis.

TaskID	Same as above
ProvenanceProbeFileID	The ID of the provenance probe, e.g. MFC2019_1518
ProvenanceProbeFileName	The partial path name to the provenance probe file (relative to the top node of the data distribution), e.g. world/MFC2019_2909.jpg
BaseFileName	Same as above
BaseBrowserFileName	The filename of the base image on the MediBrowser server
JournalName	Same as above
JournalFileName	The partial path name to the journal file relative to the top node of the data distribution e.g. reference/prov/oof7oxgiqjprd4ou4lq75wtnvdlmwhkk.json
JournalMD5	The MD5 of the journal, e.g. 7j0bkgopmzk53wti61ypx5kdpu

The second file, <DataSet_Name>-<TaskID>-ref-node.csv, documents the images in the world dataset that are associated with the probe by relating the world images present in the world dataset to the journal node ID.

ProvenanceProbeFileID	Same as above
WorldFileID	The ID of a world image associated with the probe
WorldFileName	The partial path name to the world image (relative to the top node of the data distribution), e.g. world/MFC2019_305_3.png
JournalNodeID	The ID of the node in the journal, e.g. f11fb2eac465b4e1796b3952dbcc0a47-02CA

3.5.4 Reference Files for Event Verification Task

The event verification task has the reference file <DataSet_Name>-eventverification-ref.csv that contains the following four columns.

TaskID	Same as above
ProbeFileID	Same as above
ProbeFileName	Same as above
IsTarget	Boolean indicating if the probe is from the event, i.e. "Y" | "N"
EventName	The name of the event, e.g. boston_marthon

3.5.5 Reference Files for Camera Verification Task

Three files constitute the reference files for the camera verification task. The main reference file, following the naming convention <DataSet_Name>-camera-ref.csv, contains the following nine columns.

TaskID	Same as above
ProbeFileID	Same as above
ProbeFileName	Same as above
TrainCamID	The ID of the camera device as well as the camera sensor and modality, e.g. PAR_1234_primary_image
IsTarget	Boolean indicating if the probe is from the camera, i.e. "Y" | "N"
IsManipulated	Boolean indicating if the probe is manipulated, i.e. "Y" | "N"
ProbeMaskFileName	Same as above
BaseFileName	Same as above
JournalName	Same as above

For each probe for a given Test-Train Condition, the file <DataSet_Name>-camera-ref-probejournaljoin.csv documents the journal from which the probe came as well as the operation(s), identified by the before-operation-node and after-operation-node, referenced in the <DataSet_Name>-camera-ref-journalmask.csv. Journals may include sub-graphs that do not apply to a given probe; only entries that pertain to a given probe are in the probejournaljoin file. For videos, the column BitPlane is ignored in MFC2019.

ProbeFileID	Same as above
JournalName	Same as above
StartNodeID	Same as above
EndNodeID	Same as above
BitPlane	Same as above

The file <DataSet_Name>-camera-ref-journalmask.csv documents all masks for each operation in the journal, including manipulations not necessarily included in the probe. Each row is an operation; for localizable operations, a color is provided.

JournalName	Same as above
StartNodeID	Same as above
EndNodeID	Same as above
Operation	Same as above
Color	Same as above
Purpose	Same as above
OperationArgument	Same as above
AutoDesignName	Same as above

For video probes, there are 5 additional columns.

VideoTime	Same as above
VideoFrame	Same as above
AudioTime	Same as above
AudioSample	Same as above
FrameTimeAdjustment	Same as above

3.6 Directory Structure

The data directory provided to the performer is organized as follows (in this example, MFC2019 is given as an example of <DataSet_Name>):

```
<BaseDir>
        README.txt
        /probe
                {ImageFileName1}.jpg
                {ImageFileName2}.tif
                …
                {VideoFileName1}.avi
                {VideoFileName2}.gif
                …
        /world
                {ImageFileName1}.bmp
                {ImageFileName2}.png
                …
                {VideoFileName1}.mpg
                {VideoFileName2}.wmv
                …
        /documents
        /indexes
                MFC2019-manipulation-image-index.csv
                MFC2019-manipulation-video-index.csv
                MFC2019-splice-index.csv
                MFC2019-provenancefiltering-index.csv
                MFC2019-provenance-index.csv
                MFC2019-eventverification-index.csv
                MFC2019-cameraverification-index.csv
        /reference
                /manipulation-image
                        MFC2019-manipulation-image-ref.csv
                        MFC2019-manipulation-image-ref-journalmask.csv
                        MFC2019-manipulation-image-ref-probejournaljoin.csv
                        /mask
                                {ImageFileName1}.jp2
                                {ImageFileName2}.jp2
                                …
                /manipulation-video
                        MFC2019-manipulation-video-ref.csv
                        MFC2019-manipulation-video-ref-journalmask.csv
                        MFC2019-manipulation-video-ref-probejournaljoin.csv
                        /mask
                                {VideoFileName1}.hdf5
                                {VideoFileName2}.hdf5
                                …
```

```
/splice
        MFC2019-splice-ref.csv
        MFC2019-splice-ref-journalmask.csv
        MFC2019-splace-ref-probejournaljoin.csv
        /mask
                {ImageFileName1}.jp2
                {ImageFileName2}.jp2
                …
/provenancefiltering
        MFC2019-provenancefiltering-ref.csv
        MFC2019-provenancefilgering-ref-node.csv
/provenance
        MFC2019-provenance-ref.csv
        MFC2019-provenance-ref-node.csv
/eventverification
        MFC2019-eventverification-ref.csv
/cameraverification
        MFC2019-cameraverification-ref.csv
        MFC2019-cameraverification-ref-journalmask.csv
        MFC2019-cameraverification-ref-probejournaljoin.csv
        /mask
                {ImageFileName1}.jp2
                {ImageFileName2}.jp2
                …
```

4 System Input

For a given task, a system's input is the task index file, called <DataSet_Name>-<TaskID>-index.csv and found in the 'indexes' subdirectory. Given an index file, each row specifies a test trial. Taking the corresponding image(s) or video(s) from the 'probe' and 'world' directories as input(s), systems perform detection.

4.1 Index Files

4.1.1 Index File for Manipulation Detection Task

The manipulation detection task index files for both images and videos contain task ID, file ID of the probe, probe filename, probe width, probe height, device ID, and sensor ID of the specified manipulation detection trial. The probe width and probe height fields are the rendered width and height; for images, this takes into account the EXIF (exchangeable image file format) rotation field. The device ID is the high provenance capturing device (e.g. camera) of the probe; the sensor ID is a concatenation of device ID and whether the sensor used on the device was the primary or secondary capturing sensor. For example:

TaskID	i.e. "Manipulation"
ProbeFileID	e.g. MFC2019_9291, MFC2019_3881
ProbeFileName	e.g. probe/MFC2019_9291.gif, probe/MFC2019_3881.mp4
ProbeWidth	e.g. 883
ProbeHeight	e.g. 431
HPDeviceID	e.g. PAR9082
HPSensorID	e.g. PAR9082_secondary

4.1.2 Index File for Splice Detection Task

Each trial in the index file for the splice detection task (Section 2.1.2) consists of a probe a from probe set A and potential donor b from world set B, to detect if some or all content of probe a comes from potential donor b. Each trial, thus, is an ordered pair of images, the potential Cartesian product of two datasets: the probe dataset A and the world dataset B. The Cartesian product $A \times B$ is the set of all ordered pairs (a, b) where $a \in A$ and $b \in B$.

Therefore, the splice detection task file contains task ID, probe file ID, probe filename, probe width, probe height, donor file ID, donor filename, donor width, and donor height of the specified splice detection trial.

TaskID	i.e. "Splice"
ProbeFileID	e.g. MFC2019_1281
ProbeFileName	e.g. probe/MFC2019_1281.bmp
ProbeWidth	e.g. 1680
ProbeHeight	e.g. 1050
DonorFileID	e.g. MFC2019_6864
DonorFileName	e.g. world/MFC2019_6864.nef
DonorWidth	e.g. 1054
DonorHeight	e.g. 941

4.1.3 Index File for Provenance Filtering Task

Similar to the manipulation detection task index file, the provenance filtering task file contains task ID, probe file ID, probe filename, probe width, and probe height of the specified provenance filtering trial.

TaskID	i.e. "ProvenanceFiltering"
ProvenanceProbeFileID	e.g. MFC2019_5929
ProvenanceProbeFileName	e.g. world/MFC2019_5929.nef
ProvenanceProbeWidth	e.g. 5883
ProvenanceProbeHeight	e.g. 4730

4.1.4 Index File for Provenance Graph Building Task

Similar to the manipulation detection task index file, the provenance task file contains task ID, probe file ID, probe filename, probe width, and probe height of the specified provenance filtering trial.

TaskID	i.e. "Provenance"
ProvenanceProbeFileID	e.g. MFC2019_1592
ProvenanceProbeFileName	e.g. world/MFC2019_1592.jpg
ProvenanceProbeWidth	e.g. 1489
ProvenanceProbeHeight	e.g. 3064

4.1.5 Index File for Event Verification Task

Each trial in the index file for the event verification task (Section 2.1.5) consists of a probe a from probe set A and potential event b from event set B, to detect if the probe a comes from potential event b. Each trial, thus, is an ordered pair of images, the potential Cartesian product of two datasets: the probe dataset A and the event dataset B.

The event verification task index file contains task ID, file ID of the probe, probe filename, probe width, probe height, and name of the event of the specified event verification trial. The probe width and probe height fields are the rendered width and height; for images, this takes into account the exif rotation field. For example:

TaskID	i.e. "EventVerification"
ProbeFileID	e.g. MFC2019_1383, MFC2019_9122
ProbeFileName	e.g. probe/MFC2019_1383.jpg, probe/MFC2019_9122.mp4
ProbeWidth	e.g. 1836
ProbeHeight	e.g. 909
EventName	e.g. occupy_melbourne

4.1.6 Index File for Camera Verification Task

Each trial in the index file for the camera verification task (Section 2.1.6) consists of a probe a from probe set A and potential camera device b from camera device set B, to verify if the probe a comes from putative camera device b. Each trial, thus, is an ordered pair of images, the potential Cartesian product of two datasets: the probe dataset A and the camera device dataset B.

The camera verification task index files contain task ID, file ID of the probe, probe filename, probe width, probe height, probe file size (byte unit), and ID of the training camera of the specified camera verification trial. There is one for each train-test condition. For example:

TaskID	i.e. "CameraVerification"
ProbeFileID	e.g. MFC2019_6917, MFC2019_5989
ProbeFileName	e.g. probe/MFC2019_6917.png, probe/MFC2019_5989.mpg
ProbeWidth	e.g. 1526
ProbeHeight	e.g. 800
ProbeFileSize	e.g. 512 (Byte)
TrainCamID	e.g. JK-LU-MIX_primary_multimedia

For each train-test condition, there is also a training data file (<DataSet_Name>-camera-traindata.csv) containing the ID of the training camera, file path, and file name of each image or video in the training data.

TrainCamID	e.g. 12368_primary_multimedia
FilePath	e.g. 12368/Images/Primary/No_Lens_50
FileName	e.g. _5648753.JPG

5 System Output

In this section, the types of system outputs are defined. The MediScore package[9] 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. NIST will reject submissions that do not pass validation. The MFC2019 scoring document contains instructions for how to use the validator. NIST provides the command line tools to validate MFC2019 submission files.

[9] Available at: https://github.com/usnistgov/MediScore/

5.1 Detection System Output

5.1.1 Detection System Output File

The system output file must be a CSV file that includes the confidence score and the filename of the output mask (this can be omitted if no mask is required by the task, e.g. provenance task). The filename for the output file must follow the naming convention: <SubID>/<SubID>.csv, where <SubID> is the submission identifier as described in Appendix A.

The system output CSV file for the image MDL must follow the format below:

Col1:	ProbeFileID	e.g. MFC2019_5315
Col2:	ConfidenceScore	e.g. 0.8594
Col3:	OutputProbeMaskFileName	e.g. mask/MFC2019_5315-mask.png
Col4:	ProbeStatus	i.e. “Processed” | “NonProcessed” | “OptOutAll” | “OptOutDetection” | “OptOutLocalization” | “FailedValidation”
Col5:	ProbeOptOutPixelValue	e.g. 245

The system output CSV file for the video MDL must follow the format below:

Col1:	ProbeFileID	e.g. MFC2019_1094
Col2:	ConfidenceScore	e.g. 0.4837
Col3:	ProbeStatus	i.e. “Processed” | “NonProcessed” | “OptOutAll” | “OptOutDetection” | “OptOutLocalization” | “OptOutTemporal” | “OptOutSpatial” | “FailedValidation”
Col4:	VideoFrameSegments	e.g. $\left[[3,33],[40,46]\right]$; empty JSON interval otherwise
Col5:	AudioSampleSegments	e.g. $\left[[2,18],[36,46],[48,50]\right]$; empty JSON interval otherwise
Col6:	VideoFrameOptOutSegments	e.g. $\left[[34,35]\right]$; empty JSON interval otherwise
Col7:	OutputProbeMaskFileName	e.g. mask/MFC2019_1094.hdf5
Col8:	ProbeOptOutPixelValue	e.g. 152

Note that $x_i \leq y_i$ for any given segment $[x_i, y_i]$ and that the columns VideoMaskFileName and SpacialOptOutMask are optional for MFC2019; NIST will not evaluate these columns in MFC2019.

The system output CSV file for the SDL must follow the format below:

Col1:	ProbeFileID	e.g. MFC2019_9420
Col2:	DonorFileID	e.g. MFC2019_0056
Col3:	ConfidenceScore	e.g. 0.1532
Col4:	OutputProbeMaskFileName	e.g. mask/MFC2019_9420-mask.png
Col5:	OutputDonorMaskFileName	e.g. mask/MFC2019_0056-mask.png
Col6:	ProbeStatus	i.e. “Processed” | “NonProcessed” | “OptOutAll” | “OptOutDetection” | “OptOutLocalization” | “FailedValidation”
Col7:	DonorStatus	i.e. “Processed” | “NonProcessed” | “OptOutLocalization”
Col8:	ProbeOptOutPixelValue	e.g. 137; blank if not used
Col9:	DonorOptOutPixelValue	e.g. 52; blank if not used

5.1.1.1 CONFIDENCE SCORE

The confidence score is any real number in the range $[0, 1]$ that indicates the strength of the possibility that the probe has been manipulated. The scale of the confidence score, within the range $[0, 1]$, is arbitrary but must be consistent across all testing trials, with larger values indicating greater chance that the image or video has been manipulated. Those scores are used to generate the performance curve displaying the range of possible operating characteristics.

The confidence scores for non-processed trials and for trials in which detection was opted out must be 0 (Section 5.1.1.3). Probes or donors that fail validation will be given a confidence score of 0.

5.1.1.2 VALIDATION

The ProbeFileID column in the system output <SubID>/<SubID>.csv must be consistent with the ProbeFileID column in the <BaseDir>/indexes/<DataSet_Name>-<TaskID>-index.csv file. The row order may change, but the two ProbeFileID columns must have a one-to-one correspondence.

The value of the ConfidenceScore column in the <BaseDir>/<SubID>.csv file is any real number in the range $[0, 1]$.

5.1.1.3 PROBE/DONOR STATUS

The probe status indicates if a trial was processed or not. A status of “Processed” indicates that the probe was processed and a confidence score and mask were rendered. A status of “NonProcessed” indicates that probe was not processed due to a system failure of some kind. A status of “FailedValidation” indicates that probe failed the MediScore Validator tool and will be given a score of 0—see details in the MediScore Validator ReadMe file.

A status of “OptOutAll” (or “OptOut” for tasks without localization) indicates that the system has decided not to process the probe at all, performing neither detection nor localization; unlike the previous status, this does not indicate a system failure. A status of “OptOutDetection” indicates that the system has decided not to perform detection on the probe but did perform localization. Likewise, a status of “OptOutLocalization” indicates that the system has decided to perform detection on the probe but did not perform localization; for video probes, this indicates that the system did not perform any (temporal or spatial) localization. A status of “OptOutTemporal” indicates that the system has decided to perform detection and spatial localization on the video probe but did not perform temporal localization. A status of “OptOutSpatial” indicates that the system has decided to perform detection and temporal localization on the video probe but did not perform spatial localization.

The donor status indicates if donor in an SDL trial was processed or not. The statuses are of similar definitions to probe statuses. However, there are only four possible statuses: “Processed”, “NonProcessed”, “OptOutLocalization”, “FailedValidation”.

The confidence scores for non-processed trials and for trials in which detection was opted out must be 0.

For each mask (probe and donor), the system may designate an opt-out pixel-value, indicating non-processed regions. Pixels with this probe-specific value will not be scored.

The process for determining which trials to opt out must be documented in the system description (Appendix A-a).

5.1.2 System Detection Mask Files

The mask directory contains the system output of the mask files, defined below in Sections 5.1.2.1 and 5.1.2.2, for the MDL and SDL tasks. The directory path and mask filename use the following convention: <SubID>/mask/{MaskFileName}.png for image masks or <SubID>/mask/{MaskFileName}.hdf5 for video spatial masks, where it is optional to name the mask filenames as {ProbeFileID}-mask.png or {ProbeFileID}-mask.hdf5, respectively.

5.1.2.1 Image Mask Description

The system should output a mask image to represent the detected region(s) of the manipulation for the MDL and SDL tasks. The size of the mask image must be exactly the same size as the probe image (or world image in the case of the donor mask). The mask image must be a single channel (grey) image in PNG format. Color images and images with an alpha channel will not be evaluated. For each pixel location in the input image, the system must use a one-byte integer number between 0 and 255 to indicate whether or not that pixel has been manipulated: smaller numbers indicate a greater chance that the pixel in this location has been manipulated and vice versa. In MFC2019 both binary and grey-scale masks can be evaluated. For binary masks, the system output image's pixels only have two values: 255 (not manipulated) and 0 (manipulated). For grey-scale masks, the mask scorer will report the optimum MCC over all thresholds.

For each mask (probe and donor), the system may designate an opt-out pixel-value, indicating non-processed regions. Pixels with this probe-specific value will not be scored.

5.1.2.2 Video Mask Description

The system should output a mask file to represent the detected region(s) of the manipulation for the MDL task. The mask must be in an HDF5 file. The HDF5-path to the mask is "[starting_frame_number]/masks", where [starting_frame_number] is the 0-indexed frame number indicating where the manipulated block starts and doubles as the title of the "group" to be generated in the file, and where "masks" is the name of the "dataset" within the group, which holds the frame's pixel mask data.

Each frame of the mask must have exactly the same dimensions as the probe video's frame dimensions. Each frame of the mask must be single channel (grey). Color frames and frames with an alpha channel will not be evaluated. For each pixel location per frame in the input video frame, the system must use a one-byte integer number between 0 and 255 to indicate whether or not that pixel has been manipulated: smaller numbers indicate a great chance that the pixel in this location has been manipulated and vice versa. The manipulated blocks are treated as "white frames," or frames consisting solely of the value 255.

For each mask, the system may designate an opt-out pixel-value, indicating non-processed regions. Pixels with this probe-specific value will not be scored.

5.1.2.3 Validation Rules for Mask Files

Each MaskFileName in the system output file, <SubID>/<SubID>.csv, must exist in the '<SubID>/mask' directory and be readable as a PNG file for image masks or an HDF5 file for video spatial masks. The mask file must be as described above in Sections 5.1.2.1 and 5.1.2.2. Each MaskFileName in the system output file,

<SubID>/<SubID>.csv, must have the same size as its corresponding original image defined in the system output file.

5.2 Provenance System Output

The system output for the provenance filtering and provenance graph building tasks must follow the format below:

Col1:	ProvenanceProbeFileID	e.g. MFC2019_9917
Col2:	ProvenanceOutputFileName	e.g. jsons/MFC2019_9917.json
Col3:	ProvenanceProbeStatus	i.e. "Processed" | "NonProcessed" | "OptOut" | "FailedValidation"

For details on the provenance output JSON, see Appendix C. The definitions of the provenance probe statuses are similar to those found in Section 5.1.1.3.

5.3 Event Verification System Output

The system output for the event verification task must follow the format below:

Col1:	ProbeFileID	e.g. MFC2019_2993
Col2:	EventName	e.g. boston_marathon
Col3:	ConfidenceScore	e.g. 0.5728
Col4:	ProbeStatus	i.e. "Processed" | "NonProcessed" | "OptOut" | "FailedValidation"

5.4 Camera Verification System Output

The system output for the camera verification task must follow the format below:

Col1:	ProbeFileID	e.g. MFC2019_4650
Col2:	TrainCamID	e.g. JK-LU-MIX_primary_image
Col3:	OutputProbeMaskFileName	e.g. mask/MFC2019_4650-JK-LU-MIX_primary_image-mask.png
Col4:	ConfidenceScore	e.g. 0.5485
Col5:	ProbeStatus	i.e. "Processed" | "NonProcessed" | "OptOutAll" | "OptOutDetection" | "OptOutLocalization" | "FailedValidation"
Col6:	ProbeOptOutPixelValue	e.g. 112

6 Metrics Definition for Detection Task

Two types of metrics are used in the evaluation: score metrics and mask metrics.

6.1 Score Metrics

6.1.1 Receiver Operating Characteristic (ROC)

The receiver operating characteristic (ROC) curve is used as one of the score metrics. Macmillan and Creelman [1] provide detailed information about ROC curves for detection system evaluation. Here is a brief description of the curve. In what follows, TP stands for True Positive (those correctly detected as manipulated), FN stands for False Negative (those incorrectly detected as non-manipulated), FP stands for False Positive (those incorrectly detected as manipulated), and TN stands for True Negative (those correctly detected as non-manipulated). The y-axis is the True Positive Rate (TPR) where $TPR \equiv TP/P = TP/(TP + FN)$; this is also known as sensitivity. The x-axis is the False Positive Rate (FPR) where $FPR \equiv FP/N = FP/(TN + FP) = FAR$; this is also known as 1–specificity. Figure 3 illustrates an ROC curve as the dark blue curve.

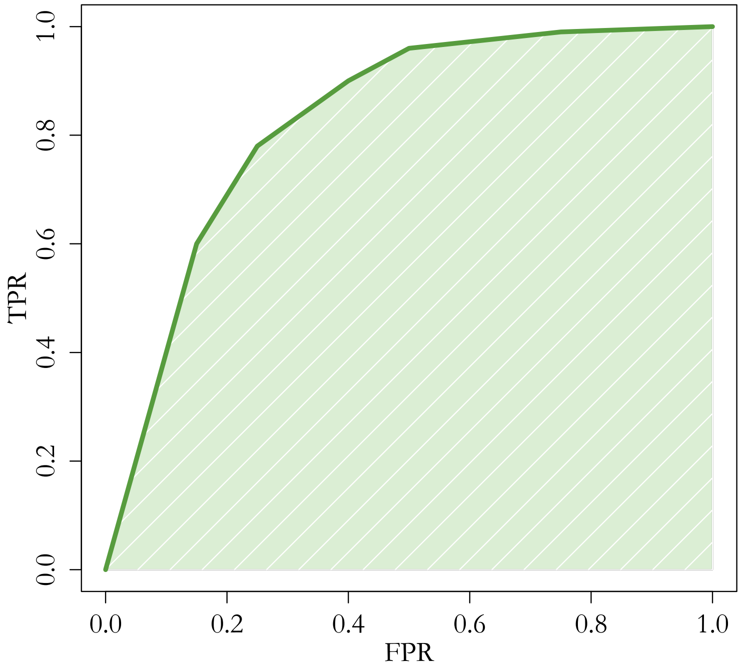

Figure 3: ROC and AUC

6.1.2 Area Under the ROC Curve (AUC)

The area under an ROC curve (AUC) is shown as the shaded beige region under the ROC curve in Figure 3. AUC quantifies the overall ability of the system to discriminate between two classes. A system no better at identifying true positives than random guessing has an AUC of 0.5. A perfect system (no false positives or false negatives) has an AUC of 1.0. The AUC-value of a system output has a value between 0 and 1.0.

6.1.3 Correct Detection (CD) at False Alarm Rate (FAR)

Another metric used is to report the Correct Detection (CD) rate on the ROC corresponding to a specified False Alarm Rate (FAR). Correct Detection rate is also known as True Positive Rate (TPR); False Alarm Rate is also known as False Positive Rate (FPR).

6.2 Mask Metrics

Three mask metrics are used: the NMM, the Matthews Correlation Coefficient (MCC), and the Weighted L1 Loss Metric (WL1). Below, all three are described in detail in Sections 6.2.2, 6.2.3, and 6.2.4, respectively. Masks are only evaluated on trials in which the specified manipulation occurred. If the system output mask for a trial was not deemed worthwhile and was therefore omitted, a mask score of -1 will be given for that trial. See Table 2 under Section 6.3 for an example.

6.2.1 Definition of Regions

Figure 4 shows a visualization of the different mask regions used for mask image evaluations. Figure 4-a shows the reference mask while Figure 4-d shows the system output mask. Figure 4-e shows the mask regions, explained below, with the weights shown in Figure 4-c after applying the dilation and erosion operations, Figure 4-b.

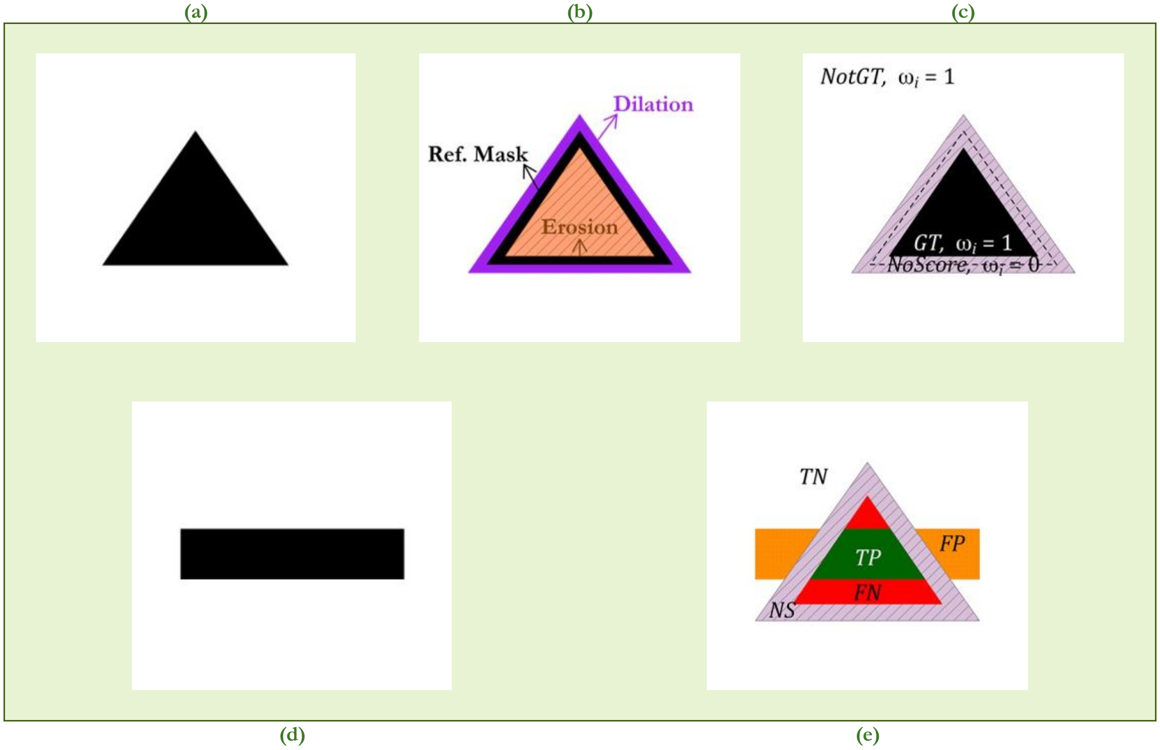

Figure 4: Mask Regions

Because of the complexity of the problem, a region around the mask will not be scored. To create this no-score region, dilation and erosion operations will be performed on the reference mask. Figure 4-b illustrates the dilation and erosion operations on the reference mask from Figure 4-a. Figure 4-c illustrates the different regions of the reference mask after the dilation and erosion operations from Figure 4-b. The solid black area in the middle, the remainder after the erosion operation, is denoted as the GT region, i.e. $GT = Erosion(M_r)$ where M_r is the black region in Figure 4-a. This is the region that will be scored as the correct manipulation region. The solid white region, the remainder after the dilation operation, is denoted as the $NotGT$ region, i.e. $NotGT = M_r - Dilation(M_r)$. This is the region that will be scored as the correct non-manipulated region. The shaded purple region between the GT and $NotGT$ regions, the result of the dilation and erosion operations, is the $NoScore$ region, i.e. $NoScore = Dilation(M_r) - Erosion(M_r)$. Any pixels in this region will be ignored for scoring purposes.

When evaluating the system output mask, Figure 4-d, using the reference mask (post dilation and erosion), Figure 4-e, the pixels are classified into the following regions based on the concepts described in [2]. Refer to Figure 4-e for all the classified regions. For spatial video masks, these regions will be taken across all frames in an interval of manipulated frames.

- True Positive (TP, also called Correct Detection, CD): The reference mask indicates it is manipulated, and the system also detected it as manipulated. The region is shown in solid green.

- False Negative (FN, also called Missed Detection, MD): The reference mask indicates it is manipulated, but the system did not detect it as manipulated. The region is shown in solid red.
- False Positive (FP, also called False Alarm, FA): The reference mask indicates it is not manipulated, but the system detected it as manipulated. The region is shown in solid orange.
- True Negative (TN, also called Correct Rejection, CR): The reference mask indicates it is not manipulated, and the system also does not detect it as manipulated. The region is shown in solid white.
- No-Score (NS): The region of the reference mask not scored, the result of the dilation and erosion operations. The region is shown in cross-hatched purple.

6.2.2 NMM

The NMM, defined below, is used to measure the accuracy of a system output mask. Before applying the NMM, the masks for each trial are normalized so that the values map from the integers 0 to 255 to the set $[0,1]$. That is, given pixel $\hat{\imath}$ from mask $\widehat{M_x}$, where $\hat{\imath}$ is an integer from 0 to 255, then $i = (255 - \hat{\imath})/255 \in [0,1]$ is the corresponding pixel-value in the normalized mask M_x. In the normalized mask, a pixel-value of 1 indicates that the pixel is manipulated, and a pixel-value of 0 indicates that the pixel is not manipulated. With non-binary grey-level masks, the normalized pixel-values indicate how certain a system is that pixel was manipulated. Note that the polarity of the mapping is inverted with larger pixel values in the mask $\widehat{M_x}$ mapping to smaller pixel values in the normalized mask M_x.

After normalization, the dilation and erosion operations are applied to the reference masks. As stated previously in Section 6.2.1, given the normalized reference mask M_r, the regions scored are the GT and $NotGT$ regions. Given system output mask M_s, where $M_s(i)$ is the ith pixel of M_s, the NMM is defined as follows:

$$\text{NMM} = \max\left\{\frac{\sum_{i\in GT} M_s(i) - \sum_{i\in GT}\big(1 - M_s(i)\big) - \sum_{i\in NotGT} M_s(i)}{\text{size}(GT)}, -1\right\}$$

This can be simplified as:

$$\text{NMM} = \max\left\{\frac{\sum_{i\in GT}(2 * M_s(i) - 1) - \sum_{i\in NotGT} M_s(i)}{\text{size}(GT)}, -1\right\}$$

When the system output mask is binary, the NMM is reduced to:

$$\text{NMM} = \max\left\{\frac{\text{size}(TP) - \text{size}(FN) - \text{size}(FP)}{\text{size}(GT)}, -1\right\}$$

Refer to Figure 4 and Section 6.2.1 for the definitions of TP, FN, and FP. Some extreme cases for the NMM in the binary case are:

- The system output mask is completely aligned with the reference mask. Then, we see that $\text{size}(TP) = \text{size}(GT)$, $\text{size}(FN) = 0$, and $\text{size}(FP) = 0$. Therefore, $\text{NMM} = 1$.
- The system output mask is completely inverted from the reference mask. Then, we see that $\text{size}(TP) = 0$, $\text{size}(FN) = \text{size}(GT)$, and $\text{size}(FP) = \text{size}(NotGT)$. Therefore, $\text{NMM} = -1$.
- The system output mask detects nothing manipulated (all pixels are 0). Then, we see that $\text{size}(TP) = 0$, $\text{size}(FN) = \text{size}(GT)$, and $\text{size}(FP) = 0$. Therefore, $\text{NMM} = -1$.

- The system output mask detects everything manipulated (all pixels are 1). Then, we see that $\text{size}(TP) = \text{size}(GT)$, $\text{size}(FN) = 0$, and $\text{size}(FP) = \text{size}(NotGT)$. Therefore, the NMM-value depends on how $\text{size}(NotGT)$ compares to $\text{size}(GT)$.
    - If $\text{size}(NotGT) < \text{size}(GT)$, then $\text{NMM} > 0$.
    - If $\text{size}(NotGT) = \text{size}(GT)$, then $\text{NMM} = 0$.
    - If $\text{size}(NotGT) > \text{size}(GT)$, then $\text{NMM} < 0$.
    - If $\text{size}(NotGT) \geq 2 * \text{size}(GT)$, then $\text{NMM} = -1$.

The NMM is invariant to translation, rotation, resizing, and cropping (under certain conditions).

6.2.3 Matthews Correlation Coefficient (MCC)

The Matthews Correlation Coefficient (MCC) is another mask metric used. Refer to Figure 4 and Section 6.2.1 for the definitions of TN, TP, FN, and FP.

$$\text{MCC} = \frac{TP * TN - FP * FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

If the denominator is zero, then we set $\text{MCC} = 0$.

If $\text{MCC} = 1$, there is perfect correlation between the reference and system output masks. If $\text{MCC} = 0$, there is no correlation between the reference and system output masks. If $\text{MCC} = -1$, there is perfect anti-correlation between the reference and system output masks.

6.2.4 Weighted L1 Loss (WL1)

The other mask metric used is Weighted L1 Loss (WL1). Given reference mask $\widehat{M_r}$ and system output mask $\widehat{M_s}$, the metric is defined as:

$$\text{WL1}\left(\widehat{M_r}, \widehat{M_s}\right) = \frac{1}{\text{size}(GT) + size(NotGT)} \sum_{i=1}^{N} \omega_i \frac{\left|\widehat{M_r}(i) - \widehat{M_s}(i)\right|}{255}$$

Here, we have $N = \text{size}\left(\widehat{M_r}\right) = \text{size}\left(\widehat{M_s}\right)$ and

$$\omega_i = \begin{cases} 0, \text{ if } i \in Dilate(M_r) \text{and } i \notin Erosion(M_r) \\ 1, \text{ otherwise} \end{cases}$$

Both mask images, $\widehat{M_r}$ and $\widehat{M_s}$, are normalized by 255.

6.2.5 Oracle Measurements for Mask Scoring

Implicit in several mask metrics is the identification of a threshold value for which the system determines a given pixel to be modified. Systems are expected to provide a single threshold to be used for all masks; however, other thresholds using the reference data as an oracle can be used. For the MFC2019, the following names designate the rule used to determine the threshold, which separates manipulated and non-manipulated pixels. These names will be used as a prefix to the measurement name, for example "Actual MCC".

- Actual – The metric is calculated using the system-provided global threshold

- Maximum (Minimum) – The metric is calculated using a single ideal global threshold found by computing metric scores over all thresholds over all masks
- Optimum – The metric is calculated using an ideal mask-specific threshold found by computing metric scores over all thresholds

6.3 Mask Scoring Evaluation Condition

As stated earlier, if performing localization, only the masks of known manipulated images will be evaluated. If no mask image is given for a trial of a known manipulated image, an NMM score of -1 will be assigned for that trial. An example is illustrated in Table 2 below.

Table 2: An Example of Outcome of Scoring System Output Masks

Image File Name	Is Manipulated?	Confidence Score	Mask File Exists?	NMM Score
MFC2019_1753.jpg	N	0.3126	N	N/A
MFC2019_0852.png	N	0.7305	Y	N/A
MFC2019_3947.png	N	0.2546	N	N/A
MFC2019_6224.tif	N	0.3939	N	N/A
MFC2019_1463.bmp	N	0.8453	Y	N/A
MFC2019_7703.nef	Y	0.7603	Y	0.591
MFC2019_0287.png	Y	0.7350	Y	0.864
MFC2019_3856.jpg	Y	0.1707	N	-1
MFC2019_8333.jpg	Y	0.2307	N	-1
MFC2019_5712.tif	Y	0.6041	Y	0.394

7 Metrics for Provenance Task

The vertex/edge overlap similarity metric from [3] is used to measure the accuracy of the system output provenance graph, G_s, for the provenance task. The set of nodes (or vertices) of the system output provenance graph is V_s while the set of links (or edges) is E_s. The reference graph is G_r with node set V_r and link set E_r. Then, the metrics are given below.

When looking at the overlap of nodes, the following metric is used:

$$\mathrm{sim}_{\mathrm{NO}}(G_r, G_s) = 2\frac{|V_r \cap V_s|}{|V_r| + |V_s|}$$

When looking at the overlap of links, the following metric is used:

$$\mathrm{sim}_{\mathrm{LO}}(G_r, G_s) = 2\frac{|E_r \cap E_s|}{|E_r| + |E_s|}$$

When looking at the overlap of both nodes and links, the following metric is used:

$$\mathrm{sim}_{\mathrm{NLO}}(G_r, G_s) = 2\frac{|V_r \cap V_s| + |E_r \cap E_s|}{|V_r| + |V_s| + |E_r| + |E_s|}$$

If $G_s = G_r$, then $\mathrm{sim}_{\mathrm{LO}} = \mathrm{sim}_{\mathrm{NO}} = \mathrm{sim}_{\mathrm{NLO}} = 1$.

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 MFC2019 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. It must begin with 'p-' for the one and only primary system (i.e., your single best system) or with 'c-' for any contrastive systems. It must then be followed by an identifier for the system (only alpha numerical characters allowed, no spaces). For example, this string could be "p-baseline" or "c-contrast". This field is intended to differentiate between runs for the same evaluation condition. Therefore, a different SysID must be used for runs where any changes were made to a system.

<VERSION> must 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> must be omitted.

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

p-baseline_3_splice

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 OptOut Criteria

Describe, if any, the strategy used to identify a trial or pixels as being opted out. If desired, voluntarily document non-processed trials.

Section 4 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, wall clock time to index the world data set and the provenance tasks, index size for the world data set, resident memory size of the index, etc.

Section 5 Training Data and Knowledge Sources

List the resources used for system development and runtime knowledge sources beyond the provided MFC corpora.

Section 6 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 description file, described in
                                         Appendix A-a
        <SubID>.csv                      The system output file, described in Section
                                         5.1.1.
        /mask                            The system output mask directory
                {MaskFileName1}.png      The system output mask file directory,
                                         described in Section 5.1.2.1
                {MaskFileName2}.png
                …
```

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

```
p-baseline_3_splice/
        p-baseline_3_splice.txt
        p-baseline_3_splice.csv
        /mask
```

A-c TRANSMITTING SUBMISSIONS

To prepare your submission, first create the previously described file/directory structure.

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 CSV File Format Specifications

The MediFor evaluation infrastructure uses comma-separated values (CSV) formatted files with an initial field header line as the data interchange format for all textual data. The EBNF structure used by the infrastructure is as follows:

```
CSVFILE      :== <HEADER> <DATA>*
<HEADER>     :== <TEXT_STRING> {“|” <TEXT_STRING> }* <NEWLINE>
<DATA>       :== <TEXT_STRING> {“|” <TEXT_STRING> }* <NEWLINE>
```

An example of the CSV content is as follows (a table and shadow is used to align the column for visualization purposes, there is no physical space between columns before the vertical bar):

```
City           |State         |Visited       |VisitDate     |Population    |…
Akron          |Ohio          |N             |              |197633        |…
```

The first data record in the files is a header line. The header lines are required by the evaluation infrastructure and the field names for the index file and the system output file are dictated by specified tasks.

Each header and data record in the table is one line of the text file. Each field value is a column and is separated from the next value with a vertical bar.

Appendix C JSON File Formal Specifications for Provenance Output

For the provenance tasks, there must be a set of JavaScript Object Notation (JSON) files, referenced in the system output file (see Section 5.2). Each probe for each task must have its own JSON file.

For provenance filtering task, the JSON file for each probe must contain 500 filtered results including the probe itself. For the provenance graph building task, the JSON file for each probe must contain only the nodes and links of the provenance graph. The JSON schemas are located in the MediScore scoring software package, available to participants.

For each node, there must be an ID, a file name, a file ID, and a confidence score for the node. The ID (`"id"`) is defined by the performer, used to identify the node in the provenance graph. The file name (`"file"`) is the file name and path defined by NIST in the task or world index file, such as in the column ProbeFileName. The file ID (`"fileid"`) is the ID defined by NIST in the task or world index file, such as the column WorldFileID. The confidence score for the node (`"nodeConfidenceScore"`) is a numerical value determined by the system, indicating the level of confidence that the node is in the provenance graph; higher values indicate more confidence.

For each link, there must be a source node, a target node, and a confidence score for the relationship. The source node (`"source"`) is the index of the source node from the node array; this node indicates which image was the start of the current manipulation action. The target node (`"target"`) is the index of the target node from the node array; this node indicates which image was the end of the current manipulation action. The confidence scores for the relationship (`"relationshipConfidenceScore"`) is a numerical value determined by the system, indicating the level of confidence that the two nodes have this relationship in the provenance graph; higher values indicate more confidence.

Appendix D DETECTION SCORER USAGE

The DetectionScorer script calculates the performance measures of AUC (see Section 6.1.2) and equal error rate (EER) based on a system's output (e.g., confidence scores) for the manipulation and splice detection tasks. Two files are outputted. The first is a CSV file containing a report table. The report table contains the measures AUC, EER, and the confidence interval for the AUC (AUC_CI). The second output is a PDF file containing a graphical plot. The plot displays an ROC (see Section 6.1.1) from the results of the algorithm performance as well as the AUC. The AUC can be partial (up to a certain FAR value) or full (when FAR value is set to 1.00).

This script also allows the user to evaluate algorithm performance on either subsets or partitions of the dataset using specified queries. To subset/partition the scored data, the command-line options utilize Pandas' queries to produce scoring reports using the metadata (e.g., Operation | Color | Purpose | OperationArgument | …) within the reference file. The relevant options regarding the query-based evaluations are summarized below.

- **Query (-q --query):** This option allows the user to specify multiple queries. Each query filters both target and non-target trials and then processes one scoring run of the system output to generate the requested scoring report.
- **Query for Partitions (-qp --queryPartition):** This option allows the user to specify only one query. The query separates the dataset into M partitions by filtering both target and non-target trials and then processes one or multiple scoring runs of the system output to generate the requested scoring report.
- **Query for Selective Manipulations (-qm --queryManipulation):** This option allows the user to specify multiple queries. Each query restricts filtering to target trials only (while using all non-target trials) to generate the requested scoring report.

D-a TEST CASE 1: FULL SCORING

```
python DetectionScorer.py -t manipulation --refDir
../../data/test_suite/detectionScorerTests/ -r reference/MFC2019-manipulation-
reference.csv -x reference/MFC2019-manipulation-index.csv --sysDir
../../data/test_suite/detectionScorerTests/baseline -s
Base_MFC2019_Manipulation_ImgOnly_p-copymove_01.csv --outRoot
./testcases/MFC19_001 --ci --display
```

Table 3: Example of Report Table Output for Test Case 1

AUC	FAR_STOP	EER	AUC_CI_LOWER	AUC_CI_UPPER
0.679533	1	0.328889	0.620826	0.735491

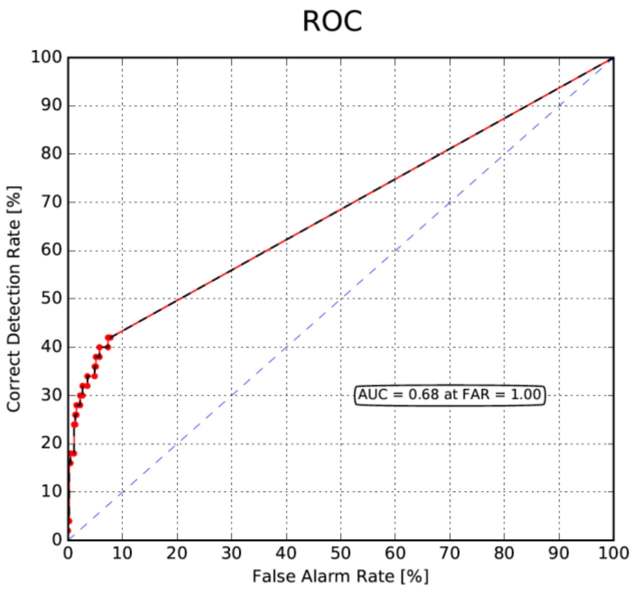

Figure 5: Example of Graphical Output for Test Case 1

D-b Test Case 2: Query (-q) with One Query

```
python DetectionScorer.py -t manipulation --refDir
../../data/test_suite/detectionScorerTests/ -r reference/MFC2019-manipulation-
reference.csv -x reference/MFC2019-manipulation-index.csv --sysDir
../../data/test_suite/detectionScorerTests/baseline -s
Base_MFC2019_Manipulation_ImgOnly_p-copymove_01.csv --outRoot
./testcases/MFC19_002 -q "Purpose ==['remove'] or IsTarget == ['N']" --ci --
display
```

Table 4: Example of Report Table Output for Test Case 2

QUERY	AUC	FAR_STOP	EER	AUC_CI_LOWER	AUC_CI_UPPER
Purpose ==['remove'] or IsTarget == ['N']	0.735463	1	0.275	0.671282	0.816155

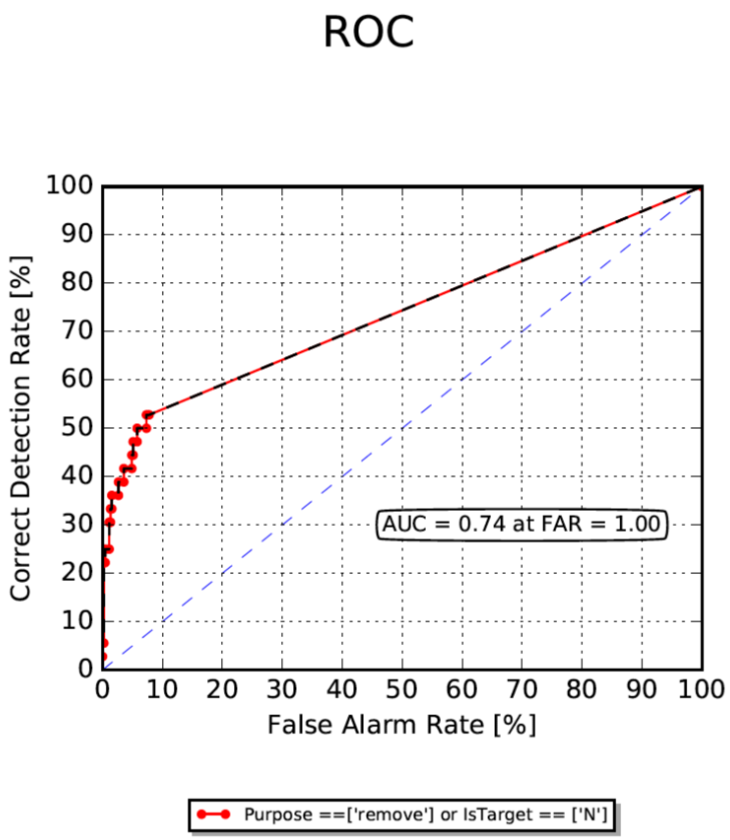

Figure 6: Example of Graphical Output for Test Case 2

D-c Test Case 3: Query for Selective Manipulation (-qm) with Two Queries

```
python DetectionScorer.py -t manipulation --refDir
../../data/test_suite/detectionScorerTests/ -r reference/MFC2019-manipulation-
reference.csv -x reference/MFC2019-manipulation-index.csv --sysDir
../../data/test_suite/detectionScorerTests/baseline -s
Base_MFC2019_Manipulation_ImgOnly_p-copymove_01.csv --outRoot
./testcases/MFC19_003 -qm "Purpose==['remove'] and Operation
==['FillContentAwareFill']" "Purpose==['remove'] and Operation
==['PasteSampled']" --ci --display
```

Table 5: Example of Report Table Output for First Query of Test Case 3

QUERY 0	AUC	FAR_STOP	EER	AUC_CI_LOWER	AUC_CI_UPPER
Purpose==['remove'] and Operation ==['FillContentAwareFill']	0.67787	1	0.330556	0.586145	0.773537

Table 6: Example of Report Table Output for Second Query of Test Case 3

QUERY 1	AUC	FAR_STOP	EER	AUC_CI_LOWER	AUC_CI_UPPER
Purpose==['remove'] and Operation ==['PasteSampled']	0.788012	1	0.223099	0.686687	0.886032

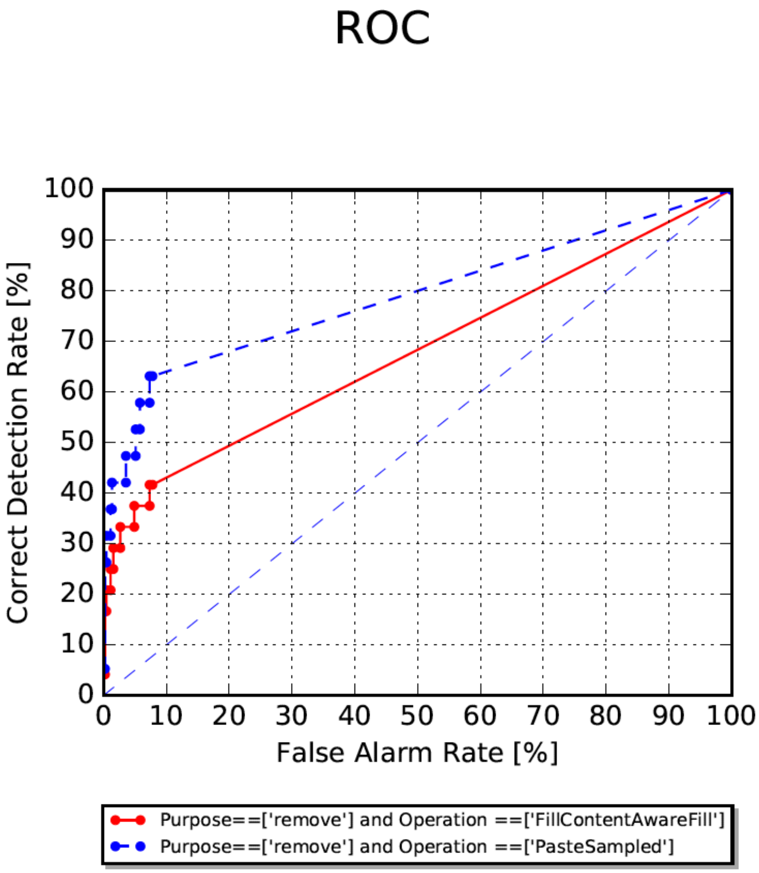

Figure 7: Example of Graphical Output for Test Case 3

REFERENCES

[1] Macmillian, N. A. & Creelman, C. D., *Detection Theory: A User's Guide*. Psychology Press, 2004.

[2] Matthews, B. W. "Comparison of the predicted and observed secondary structure of T4 phage lysozyme," *Biochim. Biophys. Acta BBA-Protein Struct.* Vol. 405. No. 2 (pp. 442-451). 1975.

[3] Papadimitriou, P., Dasdan, A. & Garcia-Molina, H. "Web graph similarity for anomaly detection," *Journal of Internet Services and Applications*. Vol. 1. No. 1(pp. 19-30). 2010.