Nimble Challenge Kickoff Dataset and Tool

Table of Contents

1 Introduction

The document descripts the kickoff dataset and tool for Media Forensics (MediFor) program (http://www.darpa.mil/program/media-forensics) sponsored by Defense Advanced Research Projects Agency (DARPA). The goal is to begin the program with a dataset and set of basic manipulations to measure baseline performance of previously published algorithms and to understand current performance in the context of a subset of potential manipulation types and quality.

The dataset and tool support Nimble Challenge 2016 (NC2016), which is a preparation for MediFor evaluations. The document covers resources, task definitions and their conditions, file formats for both system inputs and outputs, evaluation metrics, scoring procedures, and result submission protocols.

2 Tasks and Conditions

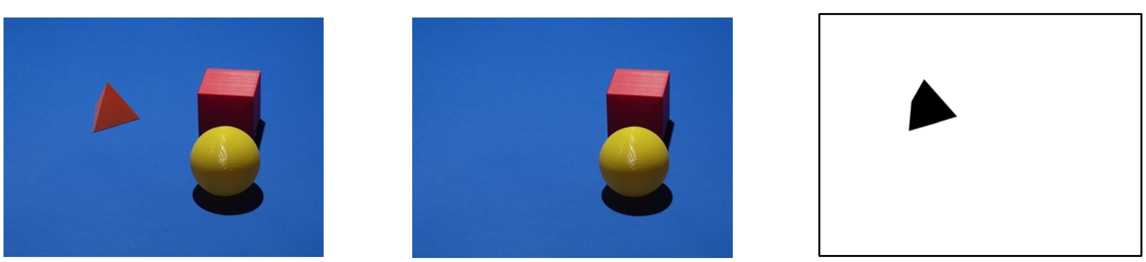
In NC2016 evaluation, three system tasks are defined: Manipulation, Removal, and Splice.

2.1 Image Manipulation Tasks

2.1.1 <u>Manipulation Detection Task (TASKID: Manipulation)</u>

In the image manipulation detection task, the objective is to detect if a probe image has been manipulated, and provide the mask image indicating the manipulated region (see Section 5.2 for detail description). For this task, the manipulation can be of any form, including cloning, splicing, removing, etc.

Data for the test is built via a set of experimentally controlled manipulation assignments. An example of a manipulation assignment for the detection task is shown in Figure 1. The orange pyramid was spliced into the base image, shown in Figure 1-2. Figure 1-1 is the resulting manipulated image. The affected pixels in the base image are identified via the mask in Figure 1-3. In manipulation detection task, the manipulated image shown in Figure1-1 is the probe image, that is, the only system input. For the manipulation task, the image source of the "orange pyramid" is not relevant to the task. The base image shown in Figure 1-2 is not provided to the performer system. The reference mask serves as the ground-truth for the performance measurement.



1. Manipulated image
2. Base image
3. Manipulation detection reference mask

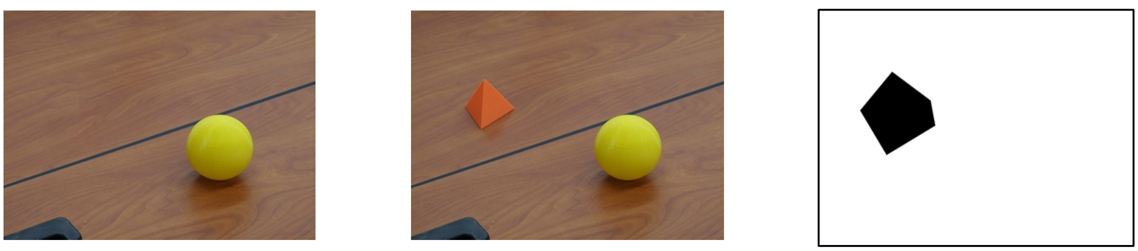
Figure 1: An example for manipulation detection task

Each manipulation trial provides a single forensic probe (Figure 1-1) for which the performer system must render a confidence score indicating the strength of evidence that the probe was manipulated as well as a system output mask, such as Figure 1-3, identifying the manipulated region(s) of the probe image. If the mask image for a trial is not worthwhile, it can be omitted.

2.1.2 Removal Detection Task (TASKID: Removal)

In the image removal detection task, the objective is to detect if an object of a forensic probe image has been removed from the image and replaced by content derived from another region within the probe image and, in addition, indicate the region that was removed. The region within the base image where the pasted-in content was acquired is not germane to the task.

An example manipulation assignment for the removal detection task is shown in Figure 2. The orange pyramid was removed from the base image, Figure 2-2, using texture derived from the same image. Figure 2-1 is the result of the removal manipulation, which is the system input (the probe image) of removal detection task. The affected pixels in the base image shown are identified via the mask image collected during the manipulation process, as shown in Figure 2-3. The reference mask image is the ground-truth for performance measurement.



1.Removal manipulation result 2.Base image 3.Reference mask image for removal detection

Figure 2: An example of an assignment for the removal detection task

Each removal manipulation assignment provides a single forensic probe (Figure 2-1) for which the system must render a confidence score indicating the strength of evidence material from the probe was removed as well as a mask, such as Figure 2-3, identifying the manipulated region(s) of the probe image. If the mask image for a trial is not worthwhile, it can be omitted.

2.1.3 Splice Detection Task (TASKID: Splice)

In the image splice detection task, the objective is to detect if a region of a splice donor image has been spliced into a probe image and, identify the region(s) in the probe image that were spliced from the donor image as well as the region(s) of the donor image that were spliced into the probe image. Note that the masks are only relevant with respect to the spliced content.

An example manipulation assignment for splice detection task is shown in Figure 3. The dandelion from the donor image, Figure 3-2, is spliced into the base image, Figure 3-3, to generate manipulated image as shown in Figure 3-1. Two mask images are collected during the manipulation procedure: the probe mask respect to the probe image as shown in Figure 3-4, and the donor mask respect to the donor image as shown in Figure 3-5.

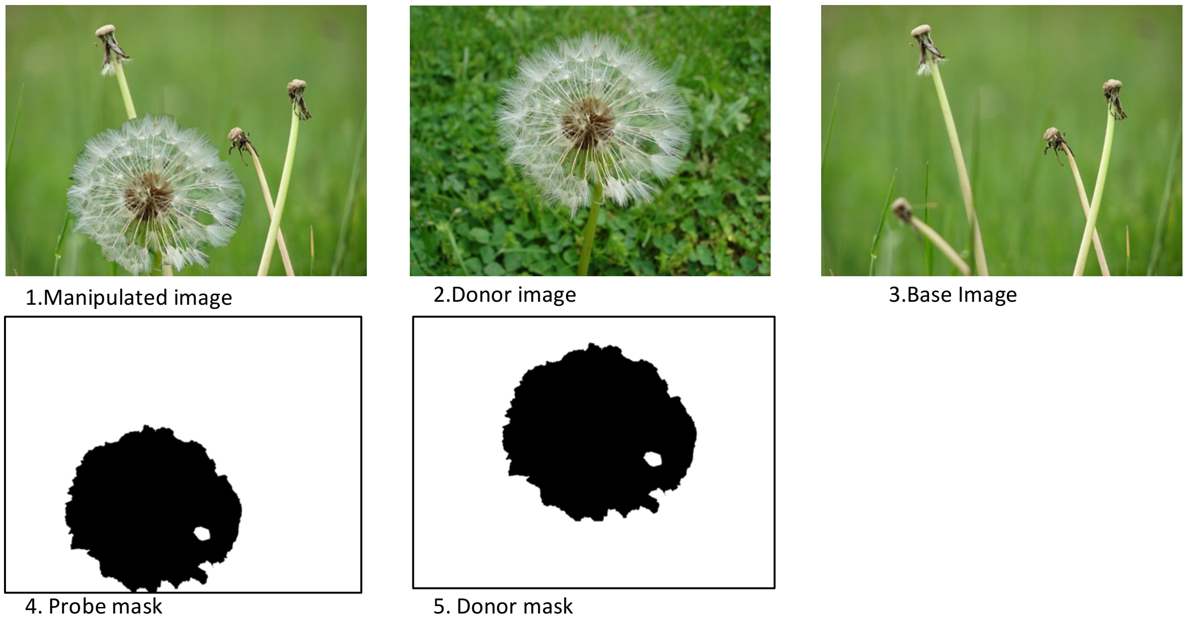


Figure 3: An example of an assignment for the splice detection task

Each splice detection trial provides a probe image, (in this case the manipulated image Figure 3-3), and a potential donor image (Figure 3-2). For each trial, the system must render a confidence score, indicating the strength of evidence that a region of the donor image was spliced into the probe image as well as a mask matching the donor mask (Figure 3-4) and a mask matching the splice manipulation mask (Figure 3-5). If both mask images are not worthwhile, they can be omitted.

2.2 Conditions

2.2.1 Image Only (CONDITIONID: ImgOnly)

For the image only condition, the performer is only allowed to use the pixel-based image content as input to their system. No image header or other information should be used.

2.2.2 Image and Metadata (CONDITIONID: ImgMeta)

The performer can use both pixel-based image content and metadata, including image header or other information as input to their system.

2.3 Protocol

All trials should be independent of each other within a given task and between tasks. Kickoff data shouldn't be used in for training. All machine learning or statistical analysis algorithms should complete training, model selection, and tuning prior to running the tests.

3 Data Resources

All data resources provided to the performer are stored in 'NC2016-Test' directory, which is the main directory for Nimble Challenge 2016. There are four main directories: 'probe', 'world', 'indexes', and 'reference'. They are explained below

3.1 Probe Directory

The NC2016 'probe' directory contains images that will be forensically analyzed. The images may be either manipulated images or non-manipulated images. The directory contains 1124 images (400 from Nimble-SCI and 724 from Nimble-WEB). In NC2016, all images are in JPG format.

3.2 World Directory

The NC2016 'world' image directory contains images to simulate a real world collection of images of unknown provenance. The directory may contain images used as donor images for some of the probe images. The NC2016 world image set contains 880 images (224 images from Nimble-SCI and 656 from Nimble-WEB). In NC2016, all images are in JPG format.

3.3 Indexes Directory

The NC2016 'indexes' directory contains a system input index file for each task. An index file is a CSV file which lists the images a system must process (see Section 4.1 and Appendix: CSV for details).

3.4 Reference Directory

The NC2016 'reference' directory contains a subdirectory for each evaluation task, i.e. manipulation, removal, or splice. Within each are two things: (1) the reference file that contains the reference metadata information about trial probes and (2) a subdirectory containing the reference masks.

There is also a 'mask' directory within each task directory (note: masks are a function of the task).

3.4.1 Reference File

The reference file(s) follows the naming convention NC2016-<TASK>-ref.csv. The reference file columns are the same for all three tasks however the contents of some columns are task-dependent. See the description below.

Col1:	TaskID	i.e. "Manipulation" | "Removal" | "**Splice**"
Col2:	ProbeFileID	e.g. NC2016_7025
Col3:	ProbeFileName	e.g. probe/NC2016_07025.jpg
Col4:	ProbeMaskFileName	e.g. reference/splice/mask/NC2016_5523.png
Col5:	DonorFileID	e.g. NC2016_3198
Col6:	DonorFileName	e.g. world/NC2016_3198.jpg
Col7:	DonorMaskFileName	e.g. reference/splice/mask/NC2016_8910.png
Col8:	IsTarget	i.e. **Y** | N
Col9:	ProbePostProcessed	i.e. "**rescale**" | "none"
Col10:	DonorPostProcessed	i.e. "rescale" | "**none**"
Col11:	ManipulationQuality	i.e. "**complex**" | "simple" | ""
Col12:	IsManipulationTypeRemoval	i.e. Y | **N**

Col13:	IsManipulationTypeSplice	i.e. **Y** | N
Col14:	IsManipulationTypeCopyClone	i.e. Y | **N**
Col15:	Collection	i.e. “Nimble-SCI” | “Nimble-WEB”
Col16:	BaseFileName	e.g. world/NC2016_7666.jpg
Col17:	Lighting	i.e. “same” | “different” | “”
Col18:	IsControl	i.e. Y | **N**
Col19:	CorrespondingControlFileName	e.g. “”
Col20:	SemanticConsistency	i.e. “consistent” | “**inconsistent**” | “”

Columns 16-19 are relevant for Nimble-SCI while Column 20 is relevant for Nimble-WEB.

3.4.2 Reference Mask

A reference mask is an image used to represent which regions of an image have been manipulated. The mask is generated during the image manipulation procedure and is a single channel (grey) image in uncompressed PNG format. The pixel of a reference mask image has two values: a value of 255 indicates that region is not manipulated (rendered as white) and a value of 0 means that region is manipulated (rendered as black).

Figure 4 shows a complex manipulation example where the Statue of Liberty in Figure 4-2 and the gorilla in Figure 4-6 are spliced into the base image, Figure 4-1.

3.4.2.1 Reference mask for manipulation (or removal) detection task

Because the objective of the manipulation or removal detection task is to find the regions (if any) where an image was manipulated (or removed), the reference mask for a manipulation (or removal) trial contains all regions of image where there has been any change (or removed). For example, if Figure 4-8 is the probe image for a manipulation detection trial, the reference mask should be Figure 4-10 showing where the Statue of Liberty and the gorilla were spliced in as well where the shadow of the gorilla was artificially made.

3.4.2.2 Reference masks for splice detection task

Because the objective of the splice detection task is to find the regions (if any) where one image was spliced into another, two reference masks are collected for each trial: probe mask and donor mask. The probe mask indicates where the donor region is used in the probe image in the splice operation alone. The image size of the probe mask is the same as the probe image. The donor mask indicates which region of the donor image is used for the splice. The image size of the splice mask is the same as the donor image.

Because this particular assignment has two splice operations from different images, two splice detection trials are generated

For the first trial, the Statue of Liberty splice, the probe image is Figure 4-8, and the potential donor image is Figure 4-2. The mask for the probe image is Figure 4-5, and the mask for the donor image is Figure 4-3.

For the second trial, the gorilla splice, the probe image is Figure 4-8, and the potential donor image is Figure 4-6. The mask for the probe image is Figure 4-9 (without the gorilla’s shadow), and the mask for the donor image is Figure 4-7.

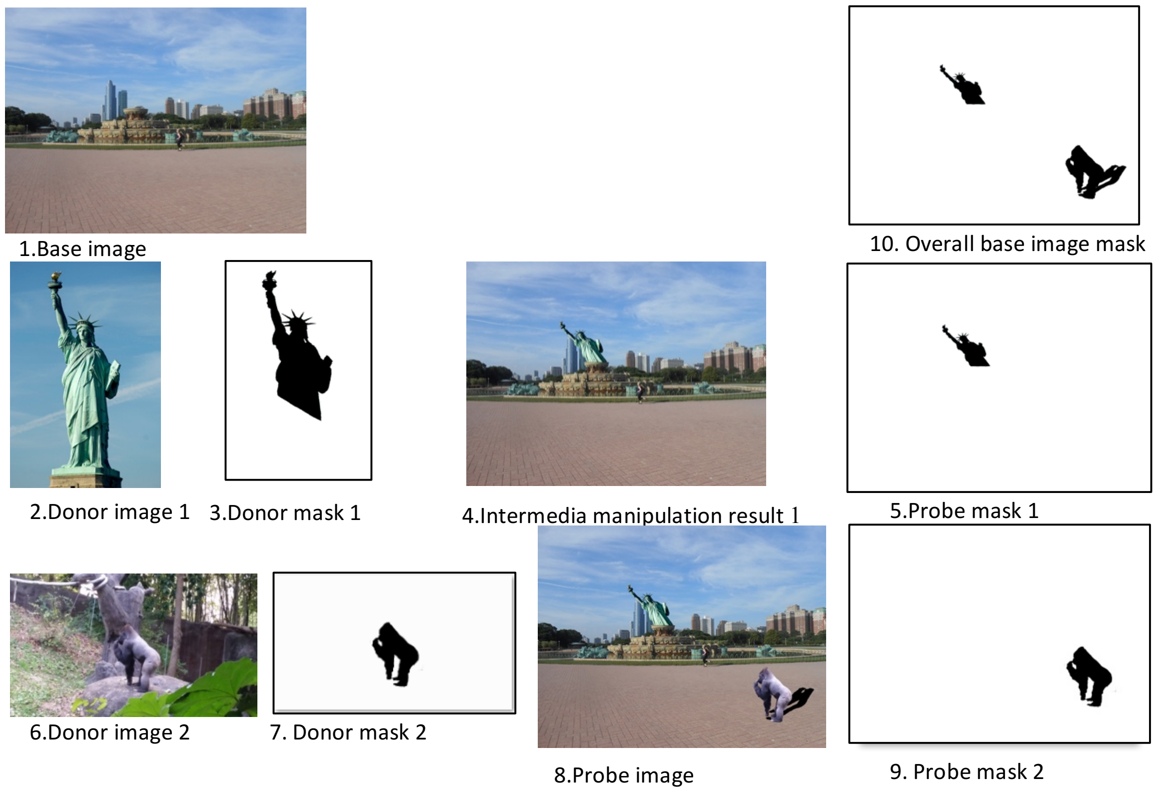


Figure 4: A multiple-splice manipulation assignment and its masks

3.5 Data Directory Structure

The data directory provided to the performer is organized as the follows:

```
NC2016-Test
    /probe
        {ImageFName1}.jpg
        {ImageFName2}.jpg
        …
        {ImageFNameN}.jpg
    /world
        {ImageFName1}.jpg
        {ImageFName2}.jpg
        …
        {ImageFNameN}.jpg
    /indexes
        NC2016-manipulation-index.csv
        NC2016-removal-index.csv
        NC2016-splice-index.csv
    /reference
        /manipulation:
                NC2016-manipulation-ref.csv
```

/mask
{ImageFName1}.png
{ImageFName2}.png
…
{ImageFNameN}.png
/removal
NC2016-removal-ref.csv
/mask
{ImageFName1}.png
{ImageFName2}.png
…
{ImageFNameN}.png
/splice
NC2016-splice-ref.csv
/mask
{ImageFName1}.png
{ImageFName2}.png
…
{ImageFNameN}.png

4 System Input

For a given task, a performer systems' input is the task index file (NC2016-<TASKID>-index.csv, defined in the following section) found in the '/indexes' subdirectory. Given an index file, each row specifies a test case. Performer systems take the corresponding image(s) from 'probe' directory and 'world' directory as the image input(s) specified for this test case and perform detection.

4.1 Index File

4.1.1 Index file for manipulation detection task

The manipulation detection task index file contains task ID, file ID of the probe image, probe filename, probe width, and probe height of the specified manipulation detection trial. For example:

Col1:	TaskID	i.e. Manipulation
Col2:	ProbeFileID	e.g. NC2016_0098
Col3:	ProbeFileName	e.g. probe/NC2016_0098.jpg
Col4:	ProbeWidth	e.g. 4032
Col5:	ProbeHeight	e.g. 3024

4.1.2 Index file for removal detection task

The index file for the removal detection task is very similar to the index file for the manipulation detection task. In the first column, TaskID, the entries will be "Removal" instead of "Manipulation."

4.1.3 Index file for splice detection task

Each test case, so called a trial, in the index file of NC2016 splice detection task (Section 2.1.3) contains an ordered pair of images from the two datasets: the probe image dataset P and the world image dataset W. These sets are described Sections 3.1 and 3.2, respectively. The Cartesian product $P \times W$ is the set of all ordered pairs (p, w) where $p \in P$ and $w \in W$.

Col1:	TaskID	i.e. "Splice"
Col2:	ProbeFileID	e.g. NC2016_0098
Col3:	ProbeFileName	e.g. probe/NC2016_0098.jpg
Col4:	ProbeWidth	e.g. 4032
Col5:	ProbeHeight	e.g. 3024
Col6:	DonorFileID	e.g. NC2016_5593
Col7:	DonorFileName	e.g. world/NC2016_5593.jpg
Col8:	DonorWidth	e.g. 5616
Col9:	DonorHeight	e.g. 3744

5 System Output

In this section, the types of system outputs are defined. The NC2016 distribution 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, as NIST will reject submissions that do not pass validation. NIST provides the command line tools to validate NC2016 submission files.

5.1 Confidence Score

The confidence score is any real number that indicates whether or not the image has been manipulated. The scale of the confidence score is arbitrary but should be consistent across all testing instances, with a smaller value indicating greater chance that the image is not manipulated and a larger value indicating greater chance that the image has been manipulated. These scores are used to generate the performance curve displaying the range of possible operating characteristics.

5.2 System Detection Mask Image

As we mentioned before, the system outputs mask image(s) to represent the detected region of the manipulation. The mask image size should be exactly the same as the input image size. The mask image should be a single channel image (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 should 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. Both binary and grey level masks are acceptable.

5.3 System Output File

5.3.1 Description

The system output file should be a spreadsheet which includes system output confidence and output mask filename, which follows the naming convention: <EXPID>/<EXPID>.csv where EXPID is the experiment identifier as described in Appendix A.1. Mask column can be omitted if no mask is required by the task.

5.3.1.1 System output file of manipulation or removal detection task

The system output file of manipulation or removal detection task follows the following format:

Col1: ProbeFileID	e.g. NC2016_0098
Col2: ConfidenceScore	e.g. 0.28
Col3: ProbeOutputMaskFileName	e.g. mask/NC2016_0098-mask.png or empty

5.3.1.2 System output file of splice detection task

The system output file of splice detection task follows the following format:

Col1: ProbeFileID	e.g. NC2016_0048
Col2: DonorFileID	e.g. NC2016_2245
Col3: ConfidenceScore	e.g. 39
Col4: ProbeOutputMaskFileName	e.g. mask/NC2016_0048-probe-mask.png or empty
Col5: DonorOutputMaskFileName	e.g. mask/NC2016_2245-donor-mask.png or empty

It is possible that probe image and donor image have the same filename of FileID, but the system output masks of probe image and donor image are different, we call such cases as copy-clone.

5.3.2 Validation Rule

- The ProbeFileID column in the system output <EXPID>/<EXPID>.csv should be consistent with the ProbeFileID column in NC2016/indexes/<EXPID>-index.csv. The row order (the order of each trial) may change, but the two ProbeFileID columns should have one-to-one correspondence.
- The DonorFileID follows the same rule for splice detection task.
- The value of ConfidenceScore column in <EXPID>/<EXPID>.csv is any real number.

5.4 System Output Mask Image

5.4.1 Description

This directory contains the system output of the image masks defined in Section 3.4.2. The directory path and mask filename use the following convention: <EXPID>/mask/{AnyName}.png, where it is optional to name mask filename as {ProbeFileID}-probe-mask.png or {DonorFileID}-donor-mask.png. For a given trial, it is acceptable that ProbeOutputMaskFileName or DonorOutputMaskFileName in system output <EXPID>/<EXPID>.csv is empty. If so, the default mask is used for evaluation, which has the same size as the probe or donor image with all pixel value equal to 255.

5.4.2 Validation Rule

- For a given trial in <EXPID>/<EXPID>.csv, if ProbeOutputMaskFileName or DonorOutputMaskFileName in system output <EXPID>/<EXPID>.csv is not empty, then the specified file should exist in the '<EXPID>/mask' directory and readable as a PNG file.
- The image of ProbeOutputMaskFileName should have the same resolution as its probe image as defined in task index file.
- The image of DonorOutputMaskFileName should have the same resolution as its donor image as defined in task index file.

5.5 System Output Directory Structure

All system output submissions must be formatted according to the following directory structure:

5.5.1 System output directory of manipulation or removal detection task

```
<EXPID>
    <EXPID>.csv
    /mask
    {ImageFName1}.png
    {ImageFName2}.png
    …
    {ImageFNameN}.png
```

5.5.2 System output directory of splice detection task

```
<EXPID>
    <EXPID>.csv
    /mask
    {ImageFName1}.png
    {ImageFName2}.png
    …
    {ImageFNameN}.png
```

Finally, zip all files and mask directories to a zip file named as <EXPID>.zip.

5.6 Tasks and System Outputs

Data files collected from performers are shown in Table 1.

Table 1: Evaluation task and performer system output

TASKID \ System Output	Confidence Score	IsTarget	Probe Mask	Donor Mask
Manipulation	✓		✓	
Removal	✓		✓	
Splice	✓	✓	✓	✓

6 Metrics for Detection Task

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

6.1 Metrics for confidence score

6.1.1 Receiver Operating Characteristic (ROC)

The ROC curve is used as one of the score metrics. Macmillan and Creelman [1] provide detailed information about ROC curves for evaluating detection system. The y-axis is True Positive Rate (TPR) where TPR ≡ TP/P; the x-axis is False Positive Rate (FPR) where FPR ≡ FP/N. Figure 5 illustrates an ROC curve.

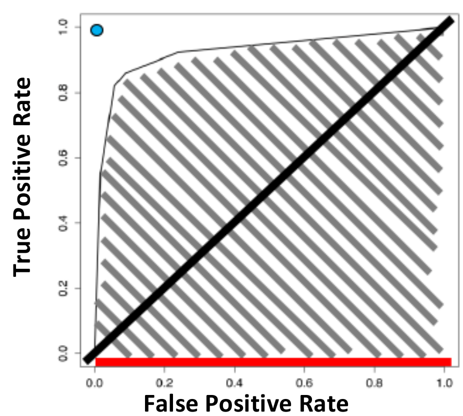


Figure 5: ROC and AUC

6.1.2 Area Under the Curve (AUC)

The area under an ROC curve (AUC) is shown as the texture region under the ROC curve in Figure 5. AUC quantifies the overall ability of the system to discriminate between two classes. AUC of a random guess system is 0.5 (the black diagonal line in Figure 5). AUC of a perfect system is 1.0 (the blue dot in Figure 5).

6.2 Metrics for system output mask

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 lowest mask score will be given for that trial.

6.2.1 Definition of regions

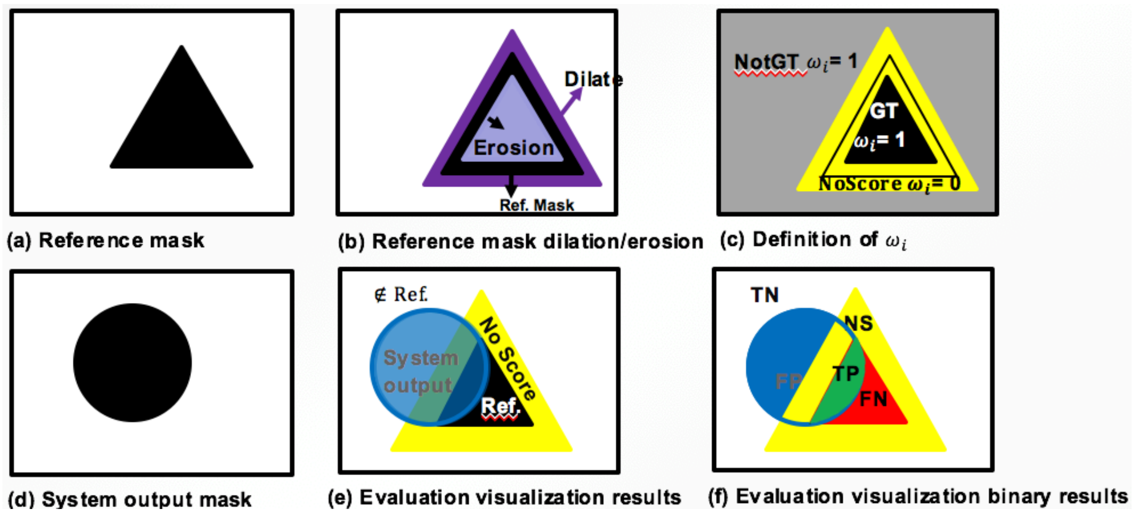


Figure 6: Mask Regions

Figure 6 shows a visualization of the different mask regions used for mask image evaluations. Figure 6-a shows a reference mask for a particular trial while Figure 6-d shows a mask outputted by a system for that trial. As defined in Section 3.4.2, the black region is the manipulated region (or the estimated manipulated region in the case of Figure 6-d).

Because of the complexity of the problem, a boundary 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 6-b illustrates the dilation and erosion operations on the reference mask from Figure 6-a. Figure 6-c illustrates the different regions of the reference mask after the dilation and erosion operations from Figure 6-b. The black area in the middle, the remainder after the erosion operation, is denoted as the GT region, i.e. $GT = Erosion(M_r)$. This is the region that will be scored as the correct manipulated region. The dark grey background 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 beige (light yellow) region in between the GT and $NotGT$ regions, the result of the dilation and erosion operations, is the no-score region. Any pixels in this region will be ignored for scoring purposes.

When evaluating the system output mask (Figure 6-d), using the reference mask (post-dilation and erosion) (Figure 6-c), the pixels are classified into the following regions based on the concepts described in [4]. Refer to Figure 6-e for all the classified regions.

- 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 green.
- False Negative (FN, also called Miss Detection, MD): The reference mask indicates it is manipulated, but the system did not detect it as manipulated. The region is shown in 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 blue.
- True Negative (TN, also called Correct Rejection, CR): The reference mask indicates it is not manipulated, and the system does not detect it as manipulated. The region is shown in white.

6.2.2 Nimble Mask Metric (NMM)

The Nimble Mask Metric (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 pixel-values indicate how certain a system is that a pixel was manipulated.

After normalization, the dilation and erosion operations are applied to the references masks. As stated previously in Section 6.2.1, given the normalized reference mask $\widehat{M_r}$, 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 6-e and Section 6.2.1 for 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, it is clear 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 the complete inversion of the reference mask. Then it is clear 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 it is clear 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 every pixel as manipulated (all pixels are 1). Then it is clear 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 NMM > 0.
- If $\text{size}(NotGT) = \text{size}(GT)$, then NMM $= 0$.
- If $\text{size}(NotGT) > \text{size}(GT)$, then NMM < 0.
- If $\text{size}(NotGT) \geq 2 * \text{size}(GT)$, then NMM $= -1$.

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

6.2.3 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:

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

where $N = \text{size}\left(\widehat{M_r}\right) = \text{size}\left(\widehat{M_s}\right)$ and

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

Both mask images, $\widehat{M_r}$ and $\widehat{M_s}$, are divided by 255. For binary system output masks, $\left|\widehat{M_r} - \widehat{M_s}\right|$ is the same concept as $\sum_{i=1}^{N} xor(r_i, s_i)$ in the definition of Hamming Loss.

Appendix A Terminology

Alpha channel – An image mask (also known as a transparency channel), where white areas define opaque color pixels, and black areas define transparent pixels. Grey levels define partial transparency, from 0 to 255, or a possible 256 shades of transparency. There are two types of masks: **Layer mask** and **Channel mask**. The layer mask represents the Alpha channel of the layer and allows you to manage its transparency. Channel mask, is also called Selection mask, which determines the transparency of a selection: by painting on a Channel Mask with white, you remove the mask and increase the selection; with black, you reduce the selection.

Exif – Exchangeable image file format is a specification for the image file format used by digital cameras. It was created by the Japan Electronic Industry Development Association (JEIDA). The specification uses the existing JPEG, TIFF Rev. 6.0, and RIFF WAVE file formats, with the addition of specific metadata tags.

Image manipulation – "The art of transforming an image to convey what you want, rather than what the original image may have shown. This can be done for artistic reasons, but because of the power of the photograph to show true depictions of reality (and the high regard that people can hold for a picture as evidence), this can also be done for reasons of deceit." [1]

Image manipulation mask – An image mask defines which region(s) of an image that has been manipulated. A white color (pixel value is 255) represents the region is manipulated, and the black color (pixel value is 0) represents the region is manipulated. Please refer to Section 3.4.2, Section 5.2, and Section 5.4 in this document for detail definition and examples.

Image resample – A process of specifying a new pixel dimension (width and height) for a whole image (global operation). However, the resample operation smooths out edges and fill in missing pixels with some interpolation methods. It is generally best to use resize on computer-generated graphics and resample on photo images.

Image resize – A process of specifying a new pixel dimension (width and height) for a whole image (global operation) without changing the 'view' of the camera/scanner.

Image splicing – A process of making a composite image by cutting and joining two or more images. The basic splice operation is to copy a region from one image and paste it to another image. In above example, the dog image is called the **donor image** and the meadow image is also called the **base image**.

Manipulation assignment – The instructions given to an image manipulator to specify what the manipulation should be and salient instructions for how to perform the manipulation. For example, given an image of a meadow with flowers and an image of a dog, the assignment is to first remove several flowers, then splice the image of the dog into the meadow image. This assignment includes two types of basic manipulation operations: removal and splice.

Scale – To enlarge or shrink a selected region of an image (local operation).

---

[1] From http://rationalwiki.org/wiki/Image_manipulation

Appendix B Instructions to the Performer

B.1 Experiment identifiers (EXPID)

The packaging and file naming conventions for NC2016 rely on **Experiment Identifiers** (EXPID) to organize and identify the files for each evaluation condition and link the system inputs to system outputs. Since EXPIDs may be used in multiple contexts, some fields contain default values. The following section describes the EXPIDs.

The following EBNF (Extended Backus–Naur Form) describes the EXPID structure with several elements:

<EXPID> ::= <TEAM>_ NC2016_ <TASK>_<CONDITION>_<SYS>_<VERSION>

Where, each element cannot contain "+" or "_" characters, and

<TEAM> ::= your short participate team name.

<TASK> ::= TASKID, which is described in Section 2.1. TASKID should be consistent with the TASKID defined in index file described in Section 4.1. In NC2016, there are only three TASKIDs: Manipulation, Removal, or Splice.

<CONDITION> ::= CONDITIONID, which is described in Section 2.2. In NC2016, there are only two CONDITIONID: ImgOnly or ImgMETA.

<SYS> ::= SYSID is the system ID. It is to begin with "p-" for the one and only primary system (i.e., your single best system) or with "c-" for any contrastive systems. It is then followed by an identifier for the system (only alphanumerical 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 should be used for runs where any changes were made to a system.

<VERSION> ::= $1 \ldots n$ (with values greater than 1 indicating multiple runs of the same experiment/system)

In order to facilitate transmission to NIST and subsequent scoring, submissions must be made using the following protocol, consisting of three steps: (1) preparing a system description, (2) packaging system outputs and system descriptions, and (3) transmitting the data to NIST.

Appendix C Comma Separated Value (CSV) File Format Specifications

The MediFor evaluation infrastructure uses comma separated values formatted files with an initial field header line as the data interchange format for all textual data. The EBNF structure the infrastructure uses 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 purpose, there is no physical space between columns before the vertical bar):

```
TaskID        |FileID        |ImgFName                  |MaskFName  |isTarget  |…
Manipulation  |NC2016_0098   |probe/NC2016_0098.jpg     |           |N         |…
```

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 D Validation and Scoring Command

In main software tool package, there are several files describe how to run the tools:

(1) README.txt file in main directory

(2) /tools/DetectionScorer/DetectionScorerReadMe.html

(3) /tools/MaskScorer/MaskScorerReadMe.html

The command line examples are shown below for illustration purpose. If there is any confliction between this document and above docuemenations, please refer to the above documentation for the latest version.

D.1 How to run validator

D.1.1 Single Source Detection (SSD) Validator for manipulation and removal detection task

In local directory of the validator, run:
$ Rscript SSDValidate.r -x <index file> -s <system output>

D.1.2 Double Source Detection (DSD) Validator for splice detection task

In local directory of the validator, run:
$ Rscript DSDValidate.r -x <index file> -s <system output>

For both SSD and DSD validator, there is also a -q option to silence certain print output:
-q 1 -- silences all non-error messages
-q 0 -- silences all print output.

D.2 How to run detection scorer (CSV version)

Detection scorer is a tool for the performance measures (ROC, AUC, and EER) on confidence scores of system outputs. The detection scorer takes the TaskID, data directory, reference file, system output file, and index file etc. as inputs, and outputs the ROC curve and AUC. Here is an example:

% Rscript DetectionScorer.r -i file -t manipulation -d "/NC2016_Test0601" -r reference/manipulation/NC2016-manipulation-ref.csv -s SystemOutputs/results.csv -x indexes/NC2016-manipulation-index.csv -m all -o myreport.csv –whichplot roc -p myroc.pdf –title Manipulation Alg1 -l dashed –size 1.5 –bySet y –byPost y

D.2 How to run mask scorer (CSV version)

Mask scorer is a tool to measure the agreement between reference mask and system output mask using defined mask metrics. The mask scorer takes TaskID, reference file, system output file, and index file etc. as inputs, and outputs the mask score and mask overlap visualization images for each trial. Here is an example of how to run mask scorer:

% Rscript MaskScorer.r -i file -t manipulation -d "/NC2016_Test0601" -r reference/manipulation/NC2016-manipulation-ref.csv -s SystemOutputs/results.csv -x indexes/NC2016-manipulation-index.csv -m all -o myreport.csv –maskout y –maskpath dct_maskoutputs –bySet y –byPost y

References

[1] N. A. Macmillan and C. D. Creelman, *Detection Theory: A User's Guide*. Psychology press, 2004.

[2] A. Martin, G. Doddington, T. Kamm, M. Ordowski, and M. Przybocki, "The DET curve in assessment of detection task performance," DTIC Document, 1997.

[3] J. G. Fiscus and G. R. Doddington, "Topic detection and tracking evaluation overview," *Top. Detect. Track.*, pp. 17–31, 2002.

[4] B. W. Matthews, "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.