Manipulation Data Collection and Annotation Tool for Media Forensics

Eric Robertson[1], Haiying Guan[2], Mark Kozak[1], Yooyoung Lee[2], Amy N. Yates[2], Andrew Delgado[2], Daniel Zhou[2], Timothee Kheyrkhah[2], Jeff Smith[3], Jonathan Fiscus[2]

[1]PAR Government, [2]National Institute of Standards and Technology, [3]University of Colorado Denver

eric_robertson@partech.com, haiying.guan@nist.gov, mark_kozak@partech.com, yooyoung.lee@nist.gov, amy.yates@nist.gov, andrew.delgado@nist.gov, daniel.zhou@nist.gov, timothee.kheyrkhah@nist.gov, jeff.smith@ucdenver.edu, jonathan.fiscus@nist.gov

Abstract

With the increasing diversity and complexity of media forensics techniques, the evaluation of state-of-the-art detectors are impeded by lacking the metadata and manipulation history ground-truth. This paper presents a novel image/video manipulation Journaling Tool (JT) that automatically or semi-automatically helps a media manipulator record, or journal, the steps, methods, and tools used to manipulate media into a modified form. JT is a unified framework using a directed acyclic graph representation to support: recording the manipulation history (journal); automating the collection of operation-specific localization masks identifying the set of manipulated pixels; integrating annotations and metadata collection; and execution of automated manipulation tools to extend existing journals or automatically build new journals.

Using JT to support the 2017 and 2018 Media Forensics Challenge (MFC) evaluations, a large collection of image manipulations was assembled that included a variety of different manipulation operations across image, video, and audio. To date, the MFC's media manipulation team has collected more than 4500 human-manipulated image journals containing over 100,000 images, more than 400 manipulated video journals containing over 4,000 videos, and generated thousands of extended journals and hundreds of auto-manipulated journals.

This paper discusses the JT's design philosophy and requirements, localization mask production, automated journal construction tools, and evaluation data derivation from journals for performance evaluation of media forensics applications. JT enriches the metadata collection, provides consistent and detailed annotations, and builds scalable automation tools to produce manipulated media, which enables the research community to better understand the problem domain and the algorithm models.

1. Introduction

Media forensics is the science and practice of determining the authenticity and establishing the integrity of an audio and visual media asset [1][2][3][4][5][6][7][8] for a variety of use cases such as litigation, fraud investigation, etc. For computer researchers, media forensics is an interdisciplinary approach to detect and identify digital media alterations using forensic techniques based on computer vision, machine learning, media imaging, statistics, etc. to identify evidence (or indicators) supporting or refuting the authenticity of a media asset.

Existing media manipulation detection technologies forensically analyze media content for indicators using a variety of information sources and techniques such as the Exchangeable Image File Format (EXIF) header, camera Photo Response Non Uniformity (PRNU) model, manipulation operation (e.g. splice, copy-clone) detection, compression anomalies, and physics-based and semantic-based consistency approaches. Before the JT, there was no manipulation annotation tool capable of capturing a historical record of the manipulation and related metadata to enable the evaluation of a wide variety of technologies and specific aspects of technologies from a single journal.

1.1. Background

DARPA's Media Forensics (MediFor) program [9] brings together world-class researchers to develop technologies for the automated integrity assessment of a media in an end-to-end platform. The primary objective for the MediFor data collection and evaluation team is to create benchmark datasets that advance current technologies and drive technological developments by understanding the key factors of this domain. The data collection and media manipulation team provides various kinds of data, metadata, and annotations supporting the program evaluations, while the evaluation team designs the data collection requirements, validates the quality of the collected data, and assembles the evaluation datasets.

1.2. Related work

The media forensics and anti-forensics techniques are developing quickly in recent years, but the evaluation and analysis of state-of-the-art manipulation detectors are impeded by the diversity and complexity of the technologies and the limitations of existing datasets, which include but are not limited to: (i) lack of rich metadata (annotations) essential to systematic evaluations and analysis; (ii) missing structured representation of manipulation history reconstruction; (iii) insufficient detail to generate diverse evaluation metadata and ground-truth (e.g. image format, manipulation semantic meaning, camera information, and manipulated image masks) for specific detectors given the same manipulated media (image or video). For a summary of existing media forensics datasets, please refer to Section 2 in [10] for details.

The ultimate goal of the MediFor program is to gain a deep understanding of the performance of different technologies based on the properties of the media, their manipulations, and their relationships with each other. In order to meet this goal, the program requires a large amount of highly diverse imagery with ground-truth labels and metadata covering an enormous spectrum of media itself, and manipulation types from the diverse image editing software and tools.

A thorough understanding of algorithm performance requires analysis of multiple factors that go into the production of manipulated imagery. These factors include the steps to produce the manipulation, software used for the manipulation, parameters provided to the software during the manipulation, and anti-forensics to disguise the manipulation. A complete assessment of state-of-the-art detectors using factor analysis with detail metadata annotations provides vital information for further advancement of current technologies. However, data collection, manipulation, and annotation are all labor intensive. In our initial manipulation and annotation study, the time used to perform the manipulation (splice, clone, or remove) is nearly the same as the time used to annotate the metadata. Our objective is to develop a tool to assist manipulators annotate data efficiently and effectively.

Computer vision researchers have developed many tools to help them collect the research data and label ground-truth, such as Photostuff [11], LabelMe [12], VATIC (Video Annotation Tool from Irvine California) [13] for the VIRAT dataset, Computer Vision Annotation Tool (CVAT) [14], VGG Image Annotator [15], Scalabel [16] and BeaverDam [17] for UC Berkeley's DeepDrive project, Polygon-RNN++ [18], and Microsoft Visual Object Tagging Tool (VoTT) [19]. Google recently announced their new 'Google Cloud Video Intelligence API' [20], which uses machine learning and the cloud to automatically analyze and annotate video content. VideoTagger [21] is another annotation tool for biological study.

Different tools are designed for different applications in different research domains. The existing annotation tools are not suitable for media forensic annotation for several reasons. (1) Most existing tools label data using annotators' basic knowledge, such as labeling an object class or segment an object out given an image. But in media forensics, given an image, the annotator may not know for sure if it was manipulated, or which pixels were changed. Such information cannot be easily recovered after the manipulation was done (see Section 4.2 for explanation and example). Therefore, annotation tools that document after the media has already been processed are not suitable for media forensic annotation. (2) Existing annotation tools do not document a thorough trace of manipulation steps. In-depth traces are needed because: Different manipulation software may implement the same function differently and leave different unique and detectable artifacts; The sequences of manipulations are not necessarily interchangeable; Anti-forensics applied during the manipulation obfuscates detection indicators; detection algorithms target specific types of manipulations and residual artifacts including light changes (artificial sources), semantic discrepancies (e.g. the Eiffel Tower in New York City), and compression effects on distributions within the Fourier domain. (3) The evaluation ground-truth data differs from historical data provided by some media editing programs, such as Adobe Photoshop or GIMP's history log files, layers, or other event recording scripts in the following aspects: The data required by evaluation differs greatly from that of software logs; The log file is an incomplete representation of all manipulation steps. It misses key data items and does not delineate backtracked or undone work; some evaluation sensitive data is not recoverable from log files or software image files; The log file is loose structure presentation, not suited for media forensic evaluation purposes; Log and script files detail software specific operation names not representative of all possible operations across the growing set of manipulation software and algorithms. Thus, these software suites do not sufficiently generalize manipulation algorithms necessary for evaluation factorization.

In this paper, we present an extensive three-year design and development project to create a novel media manipulation journaling tool that automatically or semi-automatically can collect, generate, and annotate the data and detailed manipulation metadata. JT uses a graph representation to support: i) recording the manipulation history; ii) integrating data collection, annotation, and its metadata collection, and iii) generating automated media manipulations using batch processes with pre-established manipulation sequences.

2. Evaluation data collection requirements

In addition to the labor intensity of media manipulation and metadata collection and labeling, JT is designed to support a diverse array of evaluation tasks, such as manipulation detection, localization (providing the localized manipulated region of the media), and manipulation history graph reconstruction.

Each task requires a wide variety of data and metadata to support performance evaluation and analysis using a multi-factor analysis approach. The primary requirements for image manipulation detection and localization task are: Manipulation history including intermediate images, manipulation software, operations, and its metadata; Origination data including camera, lens, environment, collection time, location etc. Semantic annotation and meaning to capture the purpose of a manipulation or series of manipulations designed to achieve a specific goal. Annotations include data identifying subject matter and setting of media. Semantic metadata includes events, weather, seasons, and intended effects of manipulation such as adding shadow or lighting inconsistency; Recompression including camera emulation as well as simulated and real-world social media images given specific procedures (e.g. Facebook image upload or download) to create realistic testing data for applications; Dynamically generated reference ground-truth mask for manipulated region.

3. Journaling Tool

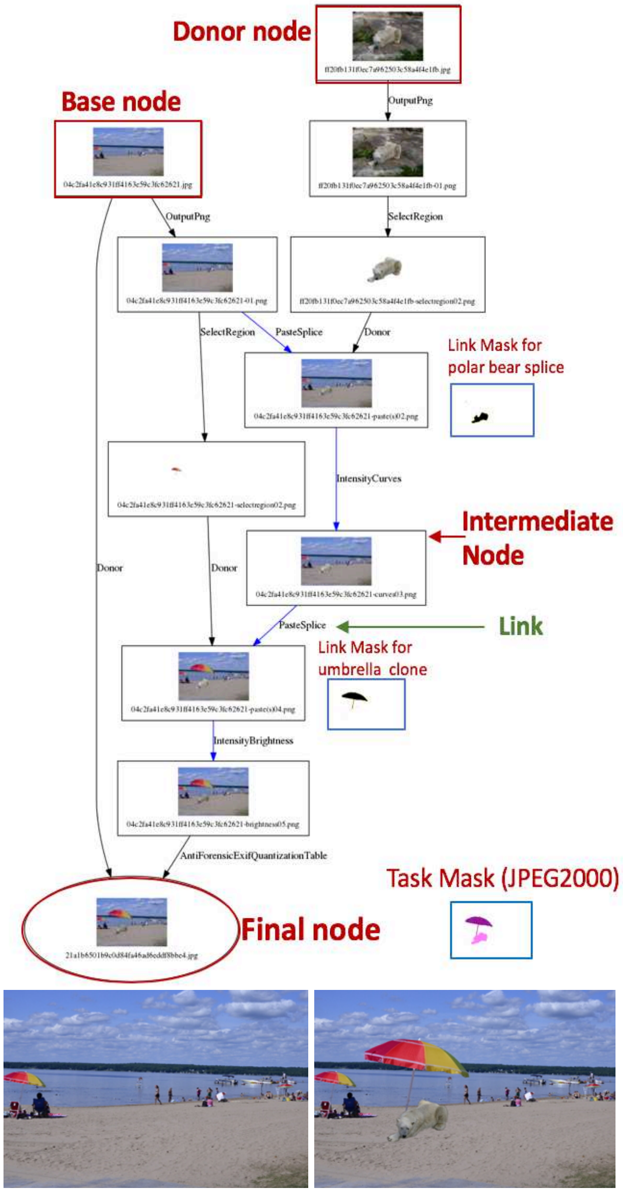
The Journaling Tool is a unified framework for data and metadata collection, annotation, and generation of automated manipulations designed according to data collection requirements. The intent of journaling is to capture a detailed history graph for each media manipulated project that results in a set of one or more final manipulated media files. The data collection process requires media manipulators to capture the detailed steps of manipulations during the manipulation process. In order to reduce the burden on manipulators, automation is built into the capture process to record incremental changes via mask generation and change analysis.

We designed a data collection approach to represent the media manipulation history with a Directed Acyclic Graph (DAG) to store manipulation history and metadata with the aim to maximize both human and machine intelligence effectively to perform multiple types of data collection and annotations. The data and metadata are collected in a hierarchical structure with three levels including a journal level, link level (a serial of operations for a given manipulated image), and node (image) level. Graph analysis algorithms support applying transformation rules to realign every image in the path from the original image to the final manipulated image, and back. The realignment provides a mapping of the original manipulation's downstream effect to the final media. The DAG structure supports reference mask generation based on different evaluation criteria.

The JT framework supports data collection and annotation throughout all stages of the manipulation process. Before manipulation begins, task design and media information are collected. During manipulation, details of each operation are captured. And upon completion post-processing produces target masks aligning operation changes to the final media product. This framework forms a comprehensive collective product we call a 'project'. During the manipulation process, details of each manipulation operation is captured that would otherwise be lost after manipulation is completed; Upon completion, post-processing produces target masks aligning operation changes to the final media products.

3.1. Manipulation history representation using DAG

In a DAG produced by JT, a node represents a media file instance such as an image, video, or audio file. An edge, referred to as a link in this paper, represents an operation that altered the source node's media to produce the destination node's media. In the general sense, the link represents a function that consumes the source and produces the destination. All metadata associated with the function is maintained with the link, including additional parameters, semantic information and change analysis. However, it is more accurate to generalize the link as a dependency between source and destination, such that the destination depends directly on the state of the source. The DAG forms a dependency tree, and, by nature of its construction, records the sequence of operations used to produce manipulated media from non-manipulated media.


Donor node
ff20fb131f0ec7a962503c58a4f4e1fb.jpg
OutputPng
Base node
04c2fa41e8c931ff4163e59c3fc62621.jpg
ff20fb131f0ec7a962503c58a4f4e1fb-01.png
OutputPng
SelectRegion
04c2fa41e8c931ff4163e59c3fc62621-01.png
ff20fb131f0ec7a962503c58a4f4e1fb-selectregion02.png
SelectRegion
PasteSplice
Donor
Link Mask for
polar bear splice
04c2fa41e8c931ff4163e59c3fc62621-paste(s)02.png
IntensityCurves
04c2fa41e8c931ff4163e59c3fc62621-selectregion02.png
Intermediate
Node
Donor
Donor
04c2fa41e8c931ff4163e59c3fc62621-curves03.png
PasteSplice
Link
Link Mask for
umbrella clone
04c2fa41e8c931ff4163e59c3fc62621-paste(s)04.png
IntensityBrightness
04c2fa41e8c931ff4163e59c3fc62621-brightness05.png
AntiForensicExifQuantizationTable
Task Mask (JPEG2000)
21a1b6501b9c0d84fa46ad6eddf8bbe4.jpg
Final node

Base node image Final node image

Figure 1: An example of a human journal (All images, graphs, and charts are original works created under contract on the MediFor Program [9]).

Figure 1 shows an example of a human manipulated journal. There are four types of nodes: base, donor, final and intermediate. A base node represents the primary media (original) being altered whereas the donor represents media contributing the alteration of the base. Base and donor nodes do not have predecessors. Base nodes represent non-manipulated, “high-provenance”, media that is camera original without any processing after capture. Donor nodes are images with un-specified provenance. Final nodes do not have successors; each final node represents a final product of a sequence of manipulations. All other interim nodes document the state of the media produced by a single manipulation.

Links form the dependencies between each manipulation state of the media. There are two kinds of links: operation and donor. An operation link represents an operation performed on a source node media file to produce a manipulated result. A donor link represents the donation of one media to the alteration of another, such as a ‘paste’ type operation would require. Although the donor is conceptually a parameter to the ‘paste’ operation, the link forms the necessary dependency.

The tool enables manipulators to record intermediate states of the media during the manipulation process, recording the incremental changes from each state to the next. Steps may be grouped to align to a semantic purpose, such as steganography.

Incremental changes include differences measured for each pixel, variations over time (video and audio), metadata alterations, and data in support of realignment of a manipulation mask to the final media including affine transform reconstruction. For example, filling a region within an image prior to a resize is portrayed as the proportionate region in the resized image.

A basic manipulation unit performs one cohesive function on the media. The operation is recorded with the name and version of the software used to perform the operation, and the parameters used in the operation. Upon completion of the operation, a mask is generated to record the specific changes made by the operation. This serves to identify the affected data of the operation and validate the operation, identifying accidental side effects. Operations are grouped into categories for ease of identification by manipulators. Paste-Splice operations involve the donation of pixels from another image. To facilitate easy recognition of the donated pixels, manipulators perform a select-region operation, applying an alpha-channel to only select donated pixels, prior to paste.

Journals are organized into patterns. These patterns set standards to avoid easily detectable side-effects. For example, images are first converted to a lossless RGB format such as Portable Network Graphics (PNG), manipulated, and then converted back to the desired media type, often applying anti-forensic operations such as re-applying specific camera quantization tables and EXIF alterations, cleansing telltale signs of manipulation.

3.2. Masks

Given a manipulated test media, its reference ground truth mask for the manipulation localization task is needed in the evaluation. The reference mask is an image where each pixel indicates whether the associated pixel in the test media has been manipulated or not. If the media was manipulated by a series of manipulations, then the reference mask is a composite mask which aggregates all manipulations’ masks along the path from a base image to the given test media in the final node. The composite mask has the same dimension with the test image and is represented in JPEG2000 format. Each channel of the JPEG2000 represents a manipulation operation mask aligned with the test image. The reference mask is aligned to the test media for uniformity over all operations including seam carving and cropping, for which the mask describes pixels removed.

In order to obtain the reference mask, up to four types of link level masks are generated: (i) Input Mask: provided by the manipulator as metadata, the mask is composed of the alpha channel of the portion of the image changed or selected, depending on the operation. The input mask is interpreted based on the operation. For Paste Clone operations, the input mask reflects the selected cloned pixels. For seam carving, the input mask reflects the protected pixels, which the manipulator does not want to change. In this case, the relative position and intensity of each pixel does not change with respect to other pixels identified in the mask.

(ii) Difference Mask: indicates the differences between the before and after manipulation operation. It was generated by capturing pixels changed during the manipulation. For the Crop operation, the difference mask reflects the change in the cropped pixels, which is expected to be none. For seam carving, the difference mask reflects the removed seams. Since full reconstruction of removed seams along two dimensions is difficult, the JT is equipped with a seam carving algorithm that records the specific seams.

(iii) Task Mask: a task-specific mask identifies the affected pixels of a final test image for a given link operation (not all link operations contribute to a task-specific mask). Global operations, such as blur and transforms, are excluded from task-specific masks. The construction of the task-specific mask includes application of all subsequent transforms to a link’s difference mask up to the final image node including, but not limited to: Resize, Rotate, Warp, Affine, Crop, Flip, Cut/Remove/Carve, and Content Aware Scale operations. In the case of seam carving where the removed seams are determined, the task mask represents those pixels neighboring removed pixels along seams.

(iv) Donor Mask: A task specific mask that identifies the donated pixels from a non-manipulated donor image. As transformations may occur prior to donation, the base image donor mask is constructed by applying antecedent transforms to the donor link’s difference mask.

The difference mask for a donor link reflects the set of pixels from the donor image pasted into an image. During the paste operation, the pasted image may be cropped, rotated and resized. Thus, the donor mask may not necessarily reflect the selected region prior to paste splice. Often the selected region from donor pixels does not represent exact pixels donated in a paste splice. In these cases, SIFT/RANSAC operation is used to determine a perspective transformation applied to the paste splice mask to produce an accurate donor mask.

3.3. JT Algorithms

JT combines human and machine intelligence to optimize the collection process. Several automatic algorithms have been developed to reduce the need for human annotation. ***Link level mask analysis***: each operation triggers operation specific analysis. All operations are concluded with structure similarity, peak signal to noise ratio and categorizations of size of change (small, medium, large). Transforms include SIFT/RANSAC computations to construct a perspective transformation matrix. Resize records the size change. Crop records both the size change and the location of the upper left pixel of the cropped area from the source image. ***Automatic single operation mask generation***: for example, local operation mask (e.g. Fill), splice paste mask, splice donor mask, and seam carving mask. ***Automatic target mask generation***: Mask generation based on the specified subtask. ***Automatic metadata generation***: journal level metadata (e.g. link count, journal complexity) and image level metadata (e.g. manipulation unit count, image complexity, manipulation summary, manipulation size).

3.4. BatchJT, AutoJT, and ExtendedJT

Evaluating specific capabilities of each detector and supporting the training of machine learning based detectors requires a good distribution and a variety of factors. Capturing detail-rich on-the-fly manipulation data adds additional burden to the manipulator, impeding the speed of producing manipulated media products.

The JT embodies three core components to automate manipulations either from start to finish or by extension of human manipulated products to quickly expand the breadth of a dataset. The first is a pipeline-based batch tool (BatchJT) to automate the creation of journals in accordance to a graph specification. The second is an automatic graph specification generating tool (AutoJT) that generates permutations of graph specifications for production of many controlled variations of journals. The third is the extended journaling tool (ExtendedJT) to automate the extension of a manipulation graph producing additional branches of manipulations off selected nodes with a set of scripted operations.

Figure 2 shows an example of an AutoJT journal (randomly generated graph). AutoJT can mimic human-based journals to produce a large number of diverse manipulation data to support statistical analysis using a wide range of manipulation types with different parameter values.

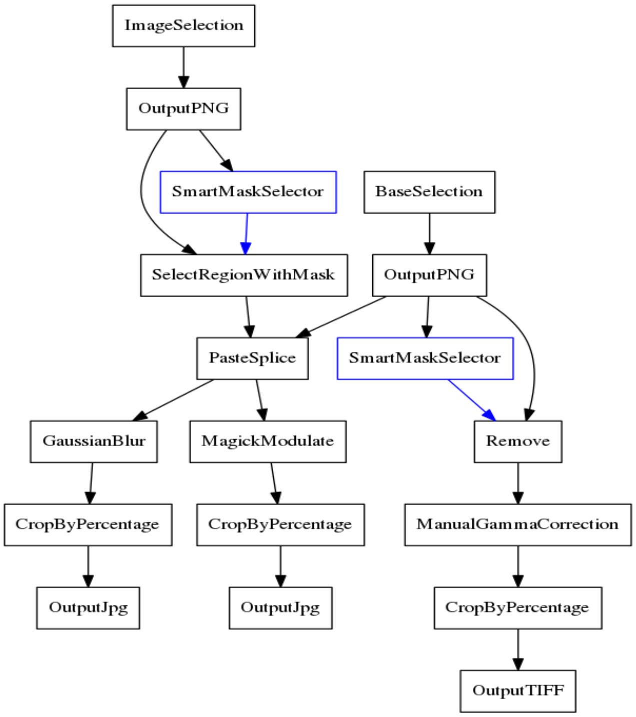


Figure 2: An example of an AutoJT journal graph

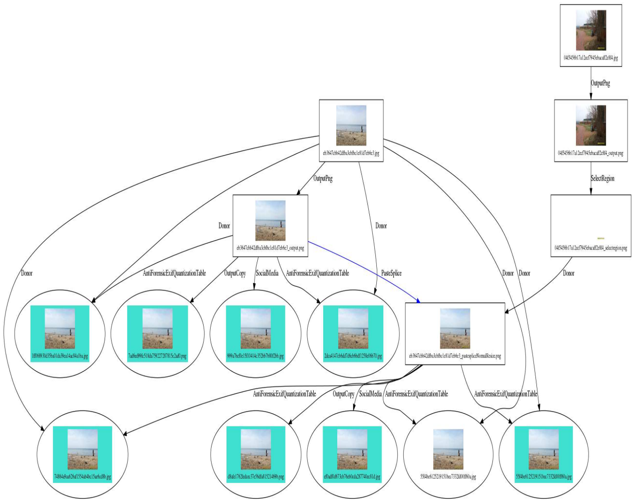

OutputPng
SelectRegion
OutputPng
Donor
Donor
Donor
Donor
Donor
Donor
AntiForensicExifQuantizationTable
OutputCopy
SocialMedia
AntiForensicExifQuantizationTable
PasteSplice
AntiForensicExifQuantizationTable
OutputCopy
SocialMedia
AntiForensicExifQuantizationTable
AntiForensicExifQuantizationTable

Figure 3: An example of an ExtendedJT journal graph.

Figure 3 shows an example of an ExtendedJT journal graph that extended a human-generated journal with four operations for each intermediate node (Facebook laundering and anti-forensics) with two different parameters, and saved in PNG format.

Through automation, permutation over the factor characteristics provide wider coverage of variability, such as evaluating crop detectors, where images are cropped with varying sizes and positions.

Furthermore, as many media transforms may be automated, an automation framework assisted by DAG specifications can direct manipulation plugins, sequenced together in controlled combinations to produce both the manipulated media and the accompanying DAGs in large batches.

3.5. JT key features

The JT generalizes mostly commonly used manipulation operations (for image, video, and audio), aligned to categories for consistent meaning across both manipulation and detection points of view.

The data and metadata collected by JT is classified into the following categories: (i) Media and supporting metadata, camera model, location, time, etc. (ii) Annotations providing semantics of individual or a sequence of manipulations; (iii) Time-of-recording manipulation masks and associated analysis data.

Metadata is organized and collected into three levels: link level, final node level, journal level. Link level captures the specific operations and change analysis of the affected medium. Final nodes capture summary information of all operations applied to a media leading to its final state. Journal properties capture the overall intent of the journal including semantics and complexity.

JT has the following major functions:

- Collect manipulation history data into a DAG;
- Generate link level evaluation masks given different types of manipulation operations;
- Generate the final image composite evaluation mask;
- Automatically generate manipulation journals given the manipulation operations and DAG graph with resources (the base image, manipulation operation and its parameters, etc.);
- Auto-Extension of existing journals;
- GAN [22] image and video journaling with GAN tools;
- Automatic video temporal frame-drops journaling for frame drop operation;
- A notification system to integrate with task management services such as project management type services and email systems;
- Validation components for quality control;
- Integrated manipulation detection tools for manipulators to assess the quality and detectability of their manipulations;
- A rich plugin architecture for adding operations, media readers (e.g. raw formats), validation rules, manipulation detection tools and remote notification systems.

3.6. JT advantages

As human manipulation on images and videos is costly and time consuming, where possible, JT reduces the burden of annotation without compromising the fidelity of the historical data. JT is a unified framework that guides the journaling process to ensure consistent and quality journals through employing the following seven concepts: (i)

Concise operation definitions for all manipulation operations along with required parameters and allowed responses. (ii) Validation rules to capture mistakes in the journaling process (e.g. resize during a format change). (iii) Quality assessment tools to ensure that donor and target masks can be aligned to donor and final image nodes, respectively, given the recorded transforms. (iv) Application of anti-forensics along with effectiveness measures to support a quantitative measure of manipulation detection difficulty. (v) ExtendedJT to quickly expand the test dataset, which uses all intermediate images generated and collected to serve as probes for different evaluation tasks. For example, one journal may contain more than 50 intermediate images associated with one final manipulated image representing the sequence of operations including blur, splice in-painting, and other transforms. Those intermediate images serve for evaluation on specific operations and the combination of those operations (e.g. splice followed by remove). (vi) Facilitate reuse and expansion of journals through extensions applied to intermediate node as required by the evaluation and/or training tasks. (vii) Automatically generate journals with a designed graph structure.

JT is publicly available as an open source package maintained on github (https://github.com/PAR-Government/media-journaling-tool). It is implemented in Python and has a detailed user guide.

4. Evaluation

4.1. Dataset generation tool: TestMaker

Given all the resources that the data collection team collected, manipulated, and annotated, the next step is to build the evaluation datasets for the task evaluations. (Please refer to [10] for all task definitions.) TestMaker is a tool to generate the evaluation test datasets with reference ground-truth data defined in [22] and used for evaluation scoring packages, MediScore, (https://github.com/usnistgov/MediScore). At the same time, TestMaker also validates journals (quality control) and metadata produced by JT and construct evaluation dataset. We will describe TestMaker in details in another document.

One of the evaluation requirements is called "selective scoring"; that is, to select a subset of data defined by a query condition (target manipulations) from the whole test pool to score a system. For the example, in Figure 1, if one would like to evaluate the performance of a Copy-Move detection system on copy-clone only images, the final image is selected as the test image, and the reference ground-truth region for evaluation is only the umbrella region.

4.2. Preliminary experiment on mask collection

As discussed in Section 1.2, post annotation of manipulated media does not capture sufficient detail to meet the needs of the evaluation program. Figure 4 demonstrates why it is important to collect and verify the mask during the manipulation process and also why most post-annotation image editing software could not provide the correct mask used for the evaluation. The first row is the original image (from the base node), the donor image (from the donor node) with a yellow ball, and the final manipulated image which pastes (or splices) the yellow ball into the first image (from the final node).

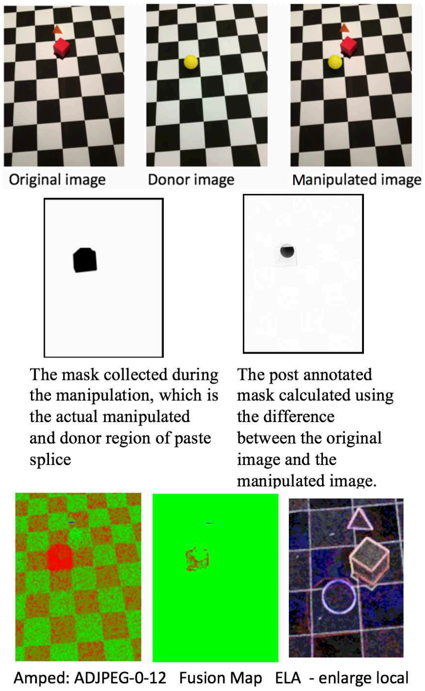
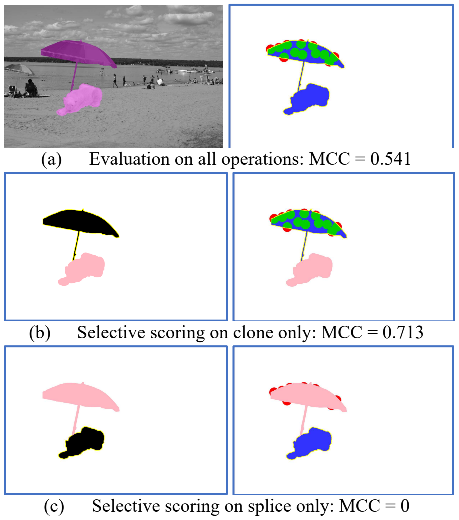

Original image
Donor image
Manipulated image
The mask collected during the manipulation, which is the actual manipulated and donor region of paste splice
The post annotated mask calculated using the difference between the original image and the manipulated image.
Amped: ADJPEG-0-12
Fusion Map
ELA - enlarge local

Figure 4: Mask collected during manipulation vs. generated after manipulation (post-annotation).

Notice that the imaging conditions of the base and donor images are very similar. The manipulator cut a polygon region from the donor image with the ball and directly pasted it to the base image. The modified pixel region should be the polygon region as shown in the first image of the second row. After the manipulation is done, there are three images: the base, donor, and final manipulated images. Using the images, one could perform post annotation to generate the mask of the manipulated region by calculating the difference of the base and manipulated images, which is the second mask of the second row with a half ball in it. This manipulation mask reflects neither the manipulated region nor its boundaries as detected using traditional detection algorithms (three algorithms in Amped Authenticate Software) as shown in the third row shows that the detection results for both region-based approach (ADJPEG) or boundary-based approach (ELA) are consistent with the mask collected during the manipulation, and are not consistent with the mask generated by post annotation. The locally enlarged ELA algorithm result shows the manipulation boundary with a red border, but the post-calculated mask cannot reflect this and so should not be used as ground-truth in the evaluation. Furthermore, such information is not captured in any image editing software and their history logs, such as Adobe Photoshop PSD files and logs. JT is designed to collect the evaluation ground-truth as required by the evaluation tasks.

4.3. Reference ground-truth masks for selective scoring

For test construction, the JT aligns manipulation masks to the evaluation media. Given a test image, the evaluation task masks for each manipulation are condensed into JPEG2000 containers in which each link mask is a bit plane. JPEG2000 is an image coding system that offers an extremely high level of scalability and accessibility. The standard supports precisions as high as 38 bits/sample. In our design, JPEG2000 was adopted to record distinct manipulations at any level for each test image, with each bit representing a distinct manipulation (represented by a distinct color in reference mask). The metadata associated with the container describes the bit plane used for each manipulation. A manipulation mask may also be associated with more than one-bit plane if the mask traverses through different transforms for two or more final manipulated media. For example, a paste splice mask may be followed by a seam carve in one evaluation media and a warp in another.



(a) Evaluation on all operations: MCC = 0.541

(b) Selective scoring on clone only: MCC = 0.713

(c) Selective scoring on splice only: MCC = 0

Figure 5: JPEG2000 mask for selective scoring evaluation

In the journal shown in Figure 1, the top beach image is the nonmanipulated base image. JT generated two local masks for the splice of the polar bear and the clone of the umbrella with their own bit plane value, expressed in the two individually colored masks in the left image of Figure 5 (a). A system output mask is shown in the right image. If the evaluation task is to detect all manipulated pixels regardless of manipulation type, then the ground-truth covers every manipulated region (all colors as shown in Figure 5 (a)). The Matthew Correlation Coefficient (MCC) of the system output mask is 0.541. If the evaluation task is to selectively evaluate only the clone detection system, then only the "clone" operation's mask should be used (the black region in the left of Figure 5 (b)) as ground-truth mask for the evaluation. The MCC of the same system output of the selective scoring on clone is 0.713. If the evaluation task is to selectively evaluate only the splice detection systems, then only "splice" operation's mask should be used (the black region in the left of Figure 5 (c)) and the selective scoring result on splice is 0.

4.4. Manipulation history graph

JT provides the accurate ground truth phylogeny graph for the evaluation of provenance building systems—those systems retrieve related images with respect to a given query image from a world dataset and construct a phylogeny graph. The world data set is composed of random images downloaded from internet and the images from the journals. To measure the accuracy of the system output, it is compared to the reference ground-truth phylogeny graph derived from the JT journal graph. Figure 6 shows an example of the evaluation results for the history graph generation system. The green boxed images are correctly retrieved nodes, green links are correctly identified links, red boxed images are incorrectly retrieved nodes, grey boxed images are non-retrieved nodes, and grey links are missing links.

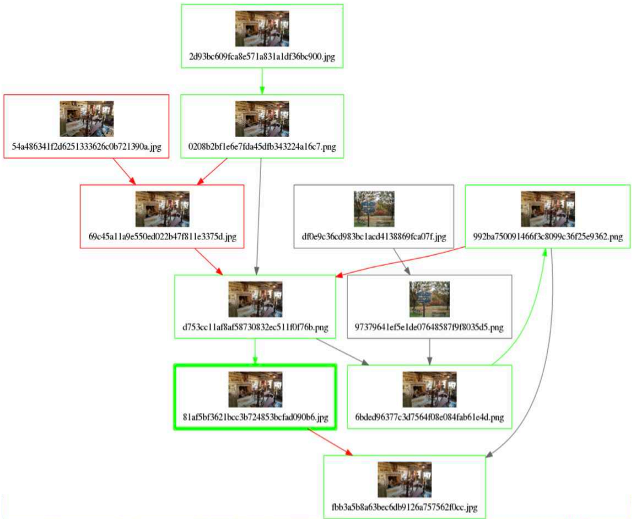

2d93bc609fca8e571a831a1df36bc900.jpg
54a486341f2d6251333626c0b721390a.jpg
0208b2bf1e6e7fda45dfb343224a16c7.png
69c45a11a9e550ed022b47f811e3375d.jpg
df0e9c36cd983bc1acd4138869fca07f.jpg
992ba750091466f3c8099c36f25e9362.png
d753cc11af8af58730832ec511f0f76b.png
97379641ef5e1de07648587f9f8035d5.png
81af5bf3621bcc3b724853bcfad090b6.jpg
6bded96377c3d7564f08e084fab61e4d.png
fbb3a5b8a63bec6db9126a757562f0cc.jpg

Figure 6: The evaluation result of a phylogeny graph produced by a provenance building system

4.5. Evaluation dataset summary

Table 1: A summary of released MediFor datasets

Image Dataset	Test Image # (K)	Journal #	Date
2017 Dev.	3.5	394	04/2017
2017 EvalPart 1	4	406	06/2017
2018 Dev1	5.6	178	12/2017
2018 Dev2	38	432	01/2018
2018 EvalPart1	17	758	03/2018

Video Dataset	Test Video #	Journal #	Date
2017 Dev.	214	23	04/2017
2017 EvalPart 1	360	47	06/2017
2018 Dev1	116	8	12/2017
2018 Dev2	231	36	01/2018
2018 EvalPart1	1036	114	03/2018

With JT, we have generated over 4500 human manipulated image journals and 400 video journals with different image and video editing programs with over 100 manipulation operations in diverse groups.

Using the manipulated image and video journals, we have generated several major evaluation datasets in the past two years. Table 1 summarizes released datasets. About 68K test images and 2K test videos with 2100 image journals and 200 video journals are available to the public. The table shows the public released dataset (one third of all evaluation data). We also have the corresponding sequestered datasets for sequester evaluation.

The metadata collected within a journal supports multi-dimensional system evaluation analysis. We can evaluate and analyze system performances by comparing across (1) different parameters for compression, image quality, resize, format, and image normalization (2) different manipulation types including splice, clone, remove and Content Aware Fill; (3) different manipulation software and algorithms including commercial off-the-shelf software, GAN, social media, etc. (4) different content type and presentations including faces, people, landscape, objects with different sizes, etc. (5) different manipulators with different skill levels and sets (6) different orders of manipulations and (7) different scanner, camera models, monitor, and printer medium, when considering recaptured media.

5. Discussion and future work

We are continuing to collect data and journals to support evaluations in future years. The design philosophy could be applied to other research domains. The JT packages are able to be adapted to other applications and purposes such as machine learning training data generation. We hope our JT framework will spur innovation in data collection and enable data-driven machine learning approaches applied to computer vision applications.

6. Acknowledgement

The authors gratefully acknowledge the members of the DARPA Media Forensics (MediFor) Program and members of the Air Force Research Lab for managing the program and weekly data team meetings; special thanks go to Matthew Turek, Neil Johnson, David Doermann, and Rajiv Jain for their instructions and strong support. The authors would like to thank Wendy Dinova-Wimmer for the initial experiments and design. PAR Government conducted this work under DARPA sponsorship via Air Force Research Laboratory (AFRL) contract FA8750-16-C-0168. NIST conducted this work under NIST Interagency Agreement Number 1505-774-08-000.

7. Disclaimer

Certain commercial equipment, instruments, software, or materials are identified in this article in order to specify the experimental procedure adequately. Such identification is not intended to imply recommendation or endorsement by NIST, nor is it intended to imply that the equipment, instruments, software or materials are necessarily the best available for the purpose.

The views, opinions and/or findings expressed are those of the author and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

References

[1] A. Piva, “An Overview on Image Forensics,” *ISRN Signal Process.*, vol. 2013, pp. 1–22, 2013.

[2] H. Farid, “Photo forensics.” The MIT Press, 2019.

[3] J. Fridrich, “Digital image forensics using sensor noise,” *Signal Proc. Mag. IEEE*, vol. 26, no. 2, pp. 26–37, 2009.

[4] J. Fridrich, “Digital image forensics,” *IEEE Signal Process. Mag.*, vol. 26, no. 2, pp. 26–37, Mar. 2009.

[5] M. Mishra and F. Adhikary, “Digital image tamper detection techniques-a comprehensive study,” *ArXiv Prepr. ArXiv13066737*, 2013.

[6] H. Farid, “Image forgery detection,” *IEEE Signal Process. Mag.*, vol. 26, no. 2, pp. 16–25, Mar. 2009.

[7] M. C. Stamm, Min Wu, and K. J. R. Liu, “Information Forensics: An Overview of the First Decade,” *IEEE Access*, vol. 1, pp. 167–200, 2013.

[8] H. T. Sencar and N. Memon, “Overview of state-of-the-art in digital image forensics,” *Algorithms Archit. Inf. Syst. Secur.*, vol. 3, pp. 325–348, 2008.

[9] M. Turek, N. Johnson etc., DARPA Media Forensics (MediFor) Program, https://www.darpa.mil/program/media-forensics.

[10] H. Guan; M. Kozak; E. Robertson; Y. Lee; A. N. Yates; A. Delgado; D. Zhou; T. Kheyrkhah; J. Smith; J. Fiscus, “MFC Datasets: Large-Scale Benchmark Datasets for Media Forensic Challenge Evaluation,” in 2019 IEEE Winter Applications of Computer Vision Workshops (WACVW), pg. 63-72, 2019.

[11] C. Halaschek-Wiener, J. Golbeck, A. Schain, M. Grove, B. Parsia, and J. Hendler, “Photostuff-an image annotation tool for the semantic web,” in Proceedings of the 4th international semantic web conference, 2005.

[12] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, “LabelMe: A Database and Web-Based Tool for Image Annotation,” Int. J. Comput. Vis., vol. 77, no. 1–3, pp. 157–173, May 2008.

[13] C. Vondrick, D. Patterson, and D. Ramanan, "Efficiently scaling up crowdsourced video annotation," International Journal of Computer Vision, vol. 101, pp. 184-204, 2013.

[14] B. Sekachev, M. Nikita, Z. Andrey et al., Computer Vision Annotation Tool (CVAT), [Online] Available: https://github.com/opencv/cvat, description link: Computer Vision Annotation Tool: A Universal Approach to Data Annotation, https://software.intel.com/en-us/articles/computer-vision-annotation-tool-a-universal-approach-to-data-annotation, March 1, 2019, [Accessed: 08-Apr-2019].

[15] A. Dutta, A. Gupta, and A. Zissermann, “VGG Image Annotator (VIA),” 2016. [Online]. Available: http://www.robots.ox.ac.uk/~vgg/software/via/. [Accessed: 08-Apr-2019].

[16] UC Berkeley DeepDrive project (https://deepdrive.berkeley.edu), “Scalabel (https://www.scalabel.ai),” 2018, [Online] Available: https://github.com/ucbdrive/scalabel. [Accessed: 08-Apr-2019].

[17] Anting Shen, “BeaverDam: Video Annotation Tool for Computer Vision Training Labels”, EECS Department, University of California, Berkeley, Master Thesis, Dec. 2016. [Online] Available: https://github.com/antingshen/BeaverDam, [Accessed: 08-Apr-2019].

[18] D. Acuna, H. Ling, A. Kar, S. Fidler, “Efficient Interactive Annotation of Segmentation Datasets with Polygon-RNN++”, 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 859 – 868, 2018.

[19] Microsoft, “VoTT: Visual Object Tagging Tool”, [Online] Available: https://github.com/Microsoft/VoTT, [Accessed: 08-Apr-2019].

[20] Google, “Cloud Video Intelligence - Video Content Analysis,” Google Cloud Platform. [Online]. Available: https://cloud.google.com/video-intelligence/. [Accessed: 29-Mar-2019]

[21] P. Rennert, O. M. Aodha, M. Piper, and G. Brostow, “VideoTagger: User-Friendly Software for Annotating Video Experiments of Any Duration,” Cold Spring Harbor Laboratory, 2018.

[22] T. Karras, T. Aila, S. Laine, and J. Lehtinen, “Progressive Growing of GANs for Improved Quality, Stability, and Variation,” International Conference on Learning Representations, 2018.

[23] A. Yates; H. Guan; Y. Lee, A. Delgado, D. Zhou, J. Fiscus, Media Forensics Challenge 2018 Evaluation Plan, NIST, 2018.