IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 27 February 2024, accepted 5 April 2024, date of publication 22 April 2024, date of current version 3 May 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3391809

RESEARCH ARTICLE

Toward Open-World Multimedia Forensics Through Media Signature Encoding

DANIELE BARACCHI[1,3], (Member, IEEE), GIULIA BOATO[2,3], FRANCESCO DE NATALE[2,3], (Senior Member, IEEE), MASSIMO IULIANI[1,3], ANDREA MONTIBELLER[2,3], CECILIA PASQUINI[4], ALESSANDRO PIVA[1,3], (Fellow, IEEE), AND DASARA SHULLANI[1,3], (Member, IEEE)

[1]Dipartimento di Ingegneria dell'Informazione (DINFO), University of Florence, 50139 Florence, Italy
[2]Dipartimento di Ingegneria e Scienza dell'Informazione (DISI), University of Trento, 38123 Trento, Italy
[3]Consorzio Nazionale Interuniversitario per le Telecomunicazioni, 43124 Parma, Italy
[4]Center for Cybersecurity, Fondazione Bruno Kessler, 38123 Trento, Italy

Corresponding author: Daniele Baracchi (daniele.baracchi@unifi.it)

This work was supported in part by the Italian Ministry of Universities and Research (MUR) under Grant 2017Z595XS, and in part by the Defense Advanced Research Projects Agency (DARPA) under Grant HR00112090136.

ABSTRACT Countering image and video manipulations is getting more and more relevant in several fields such as investigation, intelligence and forensics. Multimedia forensics researchers keep developing new tools and updating available detectors to discriminate the processing the media has been subjected to. While these tools can be utilized efficiently in controlled settings, they are generally unreliable in open-world scenarios where the investigated material may have been subjected to several unknown manipulations. In this paper, we present a novel framework to discriminate different toolchains of media manipulation and processing. We introduce the concept of media signature encoding to map image and video contents to latent spaces where media produced by similar processing toolchains cluster together. We demonstrate that this property still holds for toolchain that are not known when building the encoder, expanding the range of applications for our framework to open-world contexts where forensic analysts may face both familiar and unfamiliar manipulation techniques. A significant advantage of this approach lies in its ability to create, in principle, media signatures from any kind of forensic features. We evaluated the effectiveness of the proposed framework in two different experimental setups involving digital images and videos. Results show that encoded signatures are capable of determining whether: (i) a media under analysis belongs to a known life cycle or an entirely novel processing toolchain; (ii) a subset of media items share the same history. This framework can be considered a first step towards the use of forensic features to characterize media life cycles in open-world settings.

INDEX TERMS Multimedia forensics, media signature, feature fusion, autoencoders.

I. INTRODUCTION

Massive amounts of visual data are uploaded every day to social media platforms by nearly 4 billion active users. According to recent estimates, 14.1 billion images are shared every day and 2 million hours of video are uploaded to YouTube every minute.[1] The reason behind the popularity of sharing images and videos is actually rooted in the structure of the human brain, which is extremely fast and efficient at processing visual information as opposed to textual content. The result is that visual media are more likely to capture and hold users' attention, leading to increased engagement levels and higher sharing rates. Visual data are responsible for the viral diffusion of information through social media and web channels, and they play a key role in the digital life of individuals and societies.

At the same time, the availability of advanced tools like Artificial Intelligence (AI) and photo/video editing to the general public, which used to be restricted to skilled

The associate editor coordinating the review of this manuscript and approving it for publication was Byung-Gyu Kim.

[1]https://www.statista.com/statistics/259477/hours-of-video-uploaded-to-youtube-every-minute/

users and researchers, led to usage patterns that go beyond the primary purpose of entertainment. Deepfakes, which refer to convincing digital media that feature untruthful content,, can be obtained either through the manipulation of pristine material or generated from scratch using automated algorithms based on AI. The web abounds with tutorials and applications for the creation of simple deepfake generators that can be easily run on commercial smartphones or PCs (e.g. FakeApp, Impressions, Reface App, MyVoiceYourFace, Snapchat Cameos, FaceSwap), and more sophisticated creation techniques are developed at a fast pace. The use of AI-generated multimedia content allows marketers to cut costs and lead times for campaigns aimed at engaging customers and creating new prospects.[2]

Besides offering exciting opportunities in several fields (such as entertainment, content production, e-learning, and e-health), these advanced creation technologies are now widely recognized as a pressing threat to the reliability of visual information [1]. Indeed, malicious users and organizations have long been interested in manipulating visual contents and using them for diffusing unreliable information and fake news, especially images and videos depicting faces [2] and [3]. For instance, deepfakes are widely used for driving social engineering campaigns related to misinformation or disinformation and for obtaining financially sensitive data.[3] Moreover, research studies have shown that human performance (even of experts) in distinguishing real pictures from synthetic ones is alarmingly poor, meaning that synthesis engines have advanced beyond the uncanny valley and are now capable of creating faces that are indistinguishable or even more trustworthy than real faces [4] and [5]. This demonstrates that the detection of manipulated content and the development of tools allowing to preserve the trustworthiness of images and videos shared on social media and web platforms are important topics that our society can no longer ignore, given their significant impacts on media, public discourse, and society at large.[4]

In the latest decades multimedia forensics researchers have investigated the detection of manipulations and the identification of the source of digital content, obtaining promising results in laboratory conditions and well-defined scenarios [6], [7], [8]. Classic examples are those involving algorithms for device source attribution [9], which obtained very promising results on both images [10] and videos [11], [12], [13] when only spatial-transformations are applied, but hardly cope with complex real world conditions involving combinations of in-device image processing [14] or social-network compression [15]. Similarly, tampering and deep-fake detection algorithms [16], [17], [18] suffer from similar problems, as the specific laboratory conditions they typically consider hardly encompass the varying factors contributing to the creation of partially or fully generated data [19], [20], [21]. Although some of the assumptions made for the tests of the methods mentioned above are reasonable, the specificity of the features involved drastically reduced their application fields. Moreover, the ability of users to generate false information and deceptive content is increasing at a rapid pace, presenting significant challenges for the effectiveness of existing forensic tools in practical scenarios. The research community has recently begun efforts to expand forensic analysis to encompass real-world web-based systems, including common activities like sharing content on social media platforms [22]. However, outside of laboratory conditions the media under analysis may have possibly undergone unknown operations, and the reliability of a forensic tool should be carefully weighed. In fact, the use of forensic tools to characterize a media life cycle in open-world settings generally requires a deep knowledge of the technology behind each tool, its field of applicability, its response under unusual circumstances, and the statistical meaning of its output. As a consequence, the response of a forensic tool on contents subjected to unknown new processing can be unpredictable. These requirements make it hard to imagine how these technologies can be widely and effectively used by non expert users in the real world.

In this context, this paper presents a novel open-world multimedia forensics framework for the identification of the life cycle of a given media item under investigation. This is achieved by encoding features of different nature into a compact descriptor, called *media signature*, which is then used to quantitatively assign an object to a known class of media life cycle, and to assess whether different objects share a similar (possibly unknown) digital history. The transformation of raw features extracted from a digital content into a media signature is performed by a media-specific encoder based on a siamese-like training paradigm with denoising autoencoders, designed to preserve the traits of diverse processing chains. Therefore, the Euclidean distance between media objects in the signature space can be considered as a proxy of the similarity between the processing chains they underwent. Accordingly, we use such metric to assess the origin of a media under investigation, with respect to both known and never-seen-before processing operations. An advantage of this approach is that in principle media signatures can be generated from any kind of forensic features. As a matter of fact, we experimentally prove that the proposed methodology allows effectively encoding features extracted from both the visual content and the file structure of the object. Furthermore, the structure of the proposed media signature allows scaling the analysis to very large amounts of data. These characteristics make this framework useful to retrieve information about the life of a digital object in terms of provenance, manipulations, and sharing operations; therefore, it can support law enforcement agencies and intelligence services in tracing perpetrators of deceptive media diffusion and in countering the effects

[2]https://pavla.gr/digital-marketing-en/deepfake-technology-is-about-to-dominate-digital-marketing

[3]It is worth mentioning that the state of Texas recently passed a bill for blocking the use of deepfake to sabotage candidates during the elections, see https://legiscan.com/TX/text/SB751/id/1902830.

[4]https://www.cbinsights.com/research/report/ai-trends-2022/

of misinformation. The obtained results demonstrate the potential of the proposed approach in compelling forensic scenarios described in the next sections, as well as its ability to deal with data subject to unseen toolchains (i.e., sequence of processing operations along the media life cycle).

The paper is organized as follow: in Section II we further state the faced problem of open world forensics, the considered scenarios and precisely describe the innovations with respect to state of the art techniques; in Section III we describe the proposed framework, focusing especially on the definition, extraction and cross-indexing of media signatures; in Section IV we analyze the experimental setup on manipulated digital videos, while Section V is devoted to the experimental setup that involves shared digital images. In both cases, we define the specific descriptors and datasets, and we assess the performance of the proposed framework in the specific context. In Section VI-A, we provide a thorough discussion on the results achieved by the proposed framework, and we assess its scalability, showing that the proposed framework is capable of working on large amounts of data. Finally, in Section VII we draw the conclusions and we highlight some open issues for future works.

II. MOTIVATION AND CONTRIBUTION

The ultimate goal of image forensics is to be able to reconstruct the history of a media content by determining *a posteriori* the processing chain it went through. Current techniques are unable to attain this objective for multiple reasons. First of all, each proposed forensic detector is targeted to a specific attack and is designed and trained to reveal the corresponding traces. Second, such traces are partially erased by successive operators along the chain, thus hindering the performance of the detector. In this respect, the order of operations is also important, as different sequences usually produce very different results even when using same set of operators. Finally, when an unknown operator is introduced in the processing chain, the detector may produce unpredictable results even if the rest of the sequence is known. All these situations are very common in open-world settings, where no priors are available and media may have been shared and processed by different actors along their lifecycle.

A universal detector capable of dealing with such scenarios is currently out of reach in forensics research; however, an interesting intermediate result would be to be able to exploit the knowledge learned from previously analyzed processing chains to characterize chains using similar operations in a different order. Furthermore, when dealing with chains including never-seen operators, it would be useful to retrieve some common characteristics associated to known parts of the chain, and/or to detect similarities among media that used the same unseen operators.

The present work addresses the above ambitious objectives, by proposing a novel framework that supports the analysis of media lifecycle in open-world settings. To this purpose, we started from the basic assumption that a hard decision can rarely be achieved in an open-world scenario. Accordingly, we introduce the concept of media-signature to extract multifaceted information on the object under investigation, and quantitatively linking it to known or unknown lifecycles.

More in detail, the main novelties of the proposed approach with respect to the state of the art in the field can be summarized in the following three points:

(i) *Being able to retrieve useful information even in the presence on data subject to unseen lifecycles.* Current multimedia forensics techniques aim at discriminating fake vs. real media within a finite set of classes (e.g., by considering a set of possible manipulations, distinguishing among a finite number of possible GAN models that could have generated a fake media, or dealing with a given set of possible deepfake generators). We demonstrate that our approach is capable of retrieving useful forensic information also when the content under analysis underwent a different lifecycle with respect to training data, thus allowing us to reconstruct at least a part of the media history. For instance, when analyzing a deepfake generated by a new tool which is unknown to the classifier, the framework can recognize that it is an AI-generated video, although the specific generation tool cannot be identified.

(ii) *Clustering data with similar lifecycles.* If a completely new type of manipulation is presented, traditional forensic tools either associate the media to a random class or, when available, to a rejection class. The proposed framework takes a significant step forward. First of all, it detects whether the media belongs to a known class of manipulations or not. Second, it is able to cluster it with other unknown media that share a common history. For instance, when analyzing a set of media that have been shared multiple times over a given sequence of social networks, it could happen that (1) the social networks and the sharing sequence are already known, leading the detector to output their sharing history, or (2) the social networks and/or the sharing sequence are different from what was seen before, prompting the detector to classify the media history as unknown but, at the same time, to cluster them into the same group (meaning that they share an unknown but common history).

(iii) *Scaling to different media types and large data volumes.* Working in the real world also means being able to scale to huge amounts of data in a continuously evolving scenario. Most of the current forensics frameworks require the sequential application of different detectors, designed and trained for specific purposes, often characterized by intrinsically high complexity. This forensics framework was explicitly designed to deal with open-world scenarios, and is therefore able to encompass different types of media manipulations, to ensure effective computation, thus adapting to rapidly evolving scenarios.

The proposed framework has been extensively validated by addressing two different media forensic experimental setups, specifically designed to prove the above innovative characteristics. In the former (**media4provider**, Section IV), video sequences are analyzed to reveal the presence along the lifecycle of manipulations based on either AI

or other software-based editing operations. In the latter (**media4community**, Section V), images shared through social networks are analyzed to identify various possible sharing operations within a number of different platforms.

III. THE OPEN-WORLD FORENSIC FRAMEWORK

Given both the motivations and the requirements detailed in the previous section, the proposed framework performs a distance-based evaluation that aims at evaluating whether a media object under analysis belongs to a specific known life cycle or a previously unseen processing toolchain, thus allowing to understand whether a subset of media objects share the same history (i.e., they have undergone a similar processing/manipulation/sharing sequence of operations).

To achieve this goal we designed the architecture depicted in Figure 1. We first extract a number of features from the media under analysis (in particular, in the current implementation both content- and format-based information has been exploited); then, we encode all the extracted information into a *media signature*. The signature is a compact descriptor that maps the media under investigation into a space where it is possible to cross-index it with other media items coming from different sources, thus recognizing whether it belongs to a set of known classes of processing/manipulation/sharing or if it comes from an unknown media history.

In particular, media signatures need to be properly designed by ensure the following properties:

1) they should contain sufficient information to discriminate among media belonging to different life cycles;
2) when computed on media objects that share equal or similar life cycles, they have to be close to each other – according to a selected metric – in the signature space.

The second point is particularly relevant for real-world applications. As it is unrealistic to simulate all possible toolchains during training, our goal is to define an encoding process which can convey traces of arbitrary toolchains, so as to increase its potential in open-world settings. Typical state-of-the-art forensic detectors are in fact designed as close-set classifiers, which discriminate among toolchains that were present in the dataset used to train it. On the contrary, the proposed signature-similarity approach allows retrieving a set of similar toolchains for each given sample, as well as identifying and rejecting samples that are deemed not to belong to any of the known toolchains.

The above framework can be instantiated in several scenarios and applied to any type of media. In each case, proper feature representations will have to be determined to train the encoder and generate media signatures.

A. MEDIA SIGNATURE ENCODER

This open-world framework depends on finding an encoding function that fulfills the aforementioned properties. The search problem can be formalized as follows. Let X be the original feature space and Y be the set of all possible media life cycles, and let $f_\theta : X \to Z$ be a family of parametric functions capable of mapping the original feature vectors to a metric space $\left(Z, d : Z \times Z \to \mathbb{R}_0^+\right)$, namely, the *signature space*. Our goal is to find θ so that the distance in the signature space between samples belonging to the same class is smaller than the distance between samples belonging to different classes; in other words, we want to enforce that, given three examples $(x_1, y_1), (x_2, y_2), (x_3, y_3) \in X \times Y$ where $y_1 = y_2, y_1 \neq y_3$, we have that:

$$d(f_\theta(x_1), f_\theta(x_2)) < d(f_\theta(x_1), f_\theta(x_3)). \tag{1}$$

We adopted an approach based on machine learning, where a neural network mapping examples from $X = \mathbb{R}^n$ to $Z = \mathbb{R}^k$ is used as f_θ. Then, the definition of the encoding function becomes a supervised learning problem, where we want to estimate θ (the weights of the network) such that the relationship (1) is verified given a set of examples $(x_i, y_i) \in X \times Y$. To this purpose, we follow an approach inspired by Siamese Networks [23]. In this case, multiple examples are jointly examined, their distances evaluated, and an appropriate loss function used to force the learnt representation to meet the requirements of the similarity function. During training, for each sample x acting as an anchor, three more samples are extracted from the training set:

- a sample x_s belonging to the same class as x;
- a sample x_{n_1} belonging to a different class with respect to x;
- a sample x_{n_2} belonging to a different class with respect to both x and x_{n_1}.

These four vectors are fed into the encoder separately in order to obtain the corresponding encoded signatures $z = f_\theta(x)$, $z_s = f_\theta(x_s)$, $z_{n_1} = f_\theta(x_{n_1})$, and $z_{n_2} = f_\theta(x_{n_2})$. Finally, the parameters θ are tuned by training the network using the quadruplet loss function [24]

$$\begin{aligned} L_q = &\max\left(d(z, z_s)^2 - d(z, z_{n_1})^2 + m_1, 0\right) \\ &+ \max\left(d(z, z_s)^2 - d(z_{n_1}, z_{n_2})^2 + m_2, 0\right) \end{aligned} \tag{2}$$

where m_1 and m_2 act as regularization terms for distances among different classes, and d is the Euclidean distance on $\mathbb{R}^k$. In this way, we force the network to learn an encoding function that meets the similarity requirement in (1).

In practice, when operating in open-world scenarios the training set will include samples belonging to a (small) subset $Y^k \subset Y$ of all the existing media life cycles. As neural networks are usually trained on the assumption that the distribution of training data matches the one of test data, generalization issues arise when dealing with classes in $Y \setminus Y^k$ that are unknown at training time. In particular, our network might only retain information needed to separate classes in Y^k, while discarding cues that are useful to identify additional classes in $Y \setminus Y^k$.

To solve this problem, we enhance our siamese architecture by adding a decoding process based on Denoising Autoencoders [25]. In particular, we introduce a second network

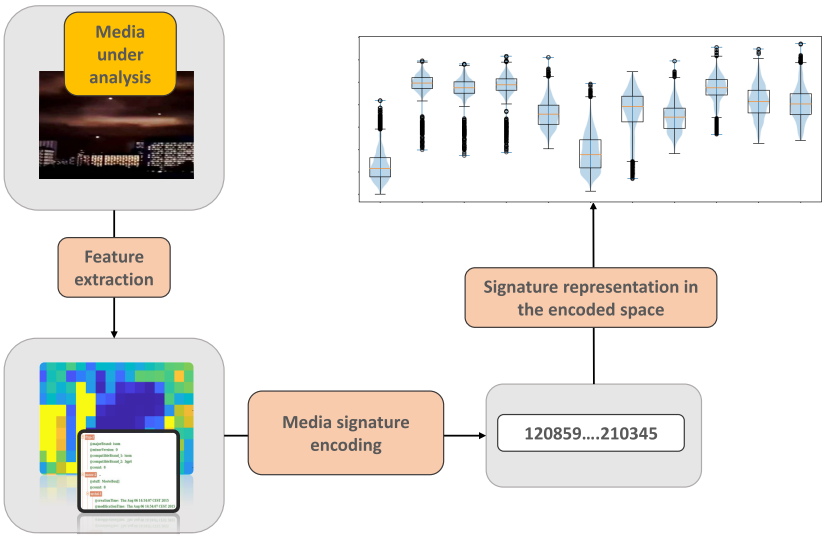


FIGURE 1. The proposed framework extracts features from a media (both content- and container-based information), encodes them into a compact descriptor (the media signature) and maps it in the signature space for forensic assessment.

$g_\psi : Z \rightarrow X$ (called decoder) designed to map vectors on the signature space back to the original feature space. This decoder is trained jointly with f_θ by minimizing the distance $|x - \hat{x}|_2^2$ between an original feature vector x and the corresponding estimate produced from its signature $\hat{x} = g_\psi(z) = g_\psi(f_\theta(x))$. This will force g_ψ to approximate the inverse function for f_θ (i.e., $g_\psi(z) \approx f_\theta^{-1}(z), \ \forall z \in Z$) and allow reconstructing x from its encoded version $f_\theta(x)$. This choice is based on the assumption that if we force the network to encode the information needed to reconstruct the original features, the signatures will keep trace of the cues that are relevant for separating unknown classes. This is formalized in the following reconstruction loss, computed on three other vectors x_s, x_{n_1}, and x_{n_2} along with each anchor x:

$$L_r = |x - \hat{x}|_2^2 + |x_s - \hat{x}_s|_2^2 + |x_{n_1} - \hat{x}_{n_1}|_2^2 + |x_{n_2} - \hat{x}_{n_2}|_2^2 . \tag{3}$$

The whole network is thus trained using a combination of the two aforementioned loss functions

$$L = \lambda_1 L_r + \lambda_2 L_q, \tag{4}$$

where the two hyperparameters λ_1 and λ_2 are used to tune the trade-off between reconstruction fidelity and separation capability. Figure 2 shows the complete architecture used for training.

It is to be noted that the decoder g_ψ is required only at training time, while the encoder f_θ will be used to extract media signatures from new examples. At test time, the system in Figure 1 takes an incoming sample, performs feature extraction, and generates the signature in the signature space, where the sample may be compared with other objects in terms of Euclidean distance. We also stress that our method makes no assumption on the shape of the original feature space. Therefore, it can be easily applied to features coming from different domains, such as discrete values extracted from the file structure (also called container) and continuous values extracted from the visual content.

In the next sections, we demonstrate the potential and flexibility of the proposed framework by instantiating it in two different experimental setups, where different media (namely videos and images) and feature representations are involved.

IV. EXPERIMENTAL SETUP 1: MEDIA4PROVIDER

In the **media4provider** setup, we consider a deceptive processing toolchain used to create a fake video to be uploaded to a web service, like a social media platform. We assume that the provider can analyze the content, before its spreading, by exploiting our framework to determine its history. We consider two main classes of manipulations (as depicted in Figure 3): (i) AI-based manipulations, including video streams where selected subjects/objects are removed and the corresponding areas are automatically generated and filled (inpainted) using last-generation AI-based techniques; (ii) user-based manipulations, including native media subjected to editing operations by means of free or commercial software for image/video manipulation (e.g., *Adobe Photoshop*, *Adobe Premiere*, *Avidemux*). In the next subsections, we discuss in detail the datasets utilized, the set of content- and container-based features employed, and the findings obtained on open-world data.

A. DATASETS

Due to technological disparities between AI-based and user-based manipulations, descriptions of instances for each class

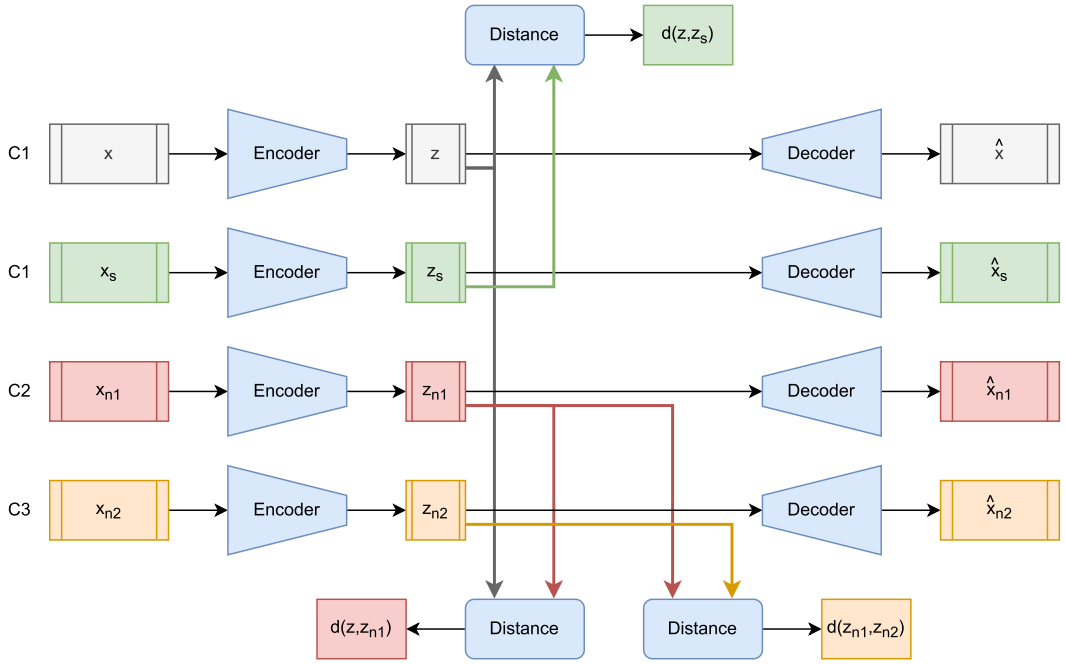


FIGURE 2. Architecture of the proposed media signature extractor. During training, for each example x (anchor) belonging to class C_1 we sample three additional vectors: x_s from the same class C_1 (in green), x_{n_1} from a different class C_2 (in red), and x_{n_2} from a third class C_3 (in orange). All of them are projected to the signature space by the encoder network, producing four signatures z, z_s, z_{n_1}, and z_{n_2}. The decoder network is then used to estimate the original features $\hat{x}$, $\hat{x}_s$, $\hat{x}_{n_1}$, and $\hat{x}_{n_2}$ from the signatures, which are then compared by the reconstruction loss. At the same time, the quadruplet loss forces signatures belonging to the same class $d(z, z_s)$ to be near each other, while maximizing the distances $d(z, z_{n_1})$ and $d(z, z_{n_2})$ of signatures belonging to different classes.

TABLE 1. Summary of the considered datasets for the Experimental Setup 1.

Dataset	Manipulation Instance	Manipulation Type
Dataset Inpainting (1248 videos)	STTN, OPN, GM-CNN	Inpainting
EVA-7k (1260 videos)	Avidemux, Adobe Premiere, Kdenlive, ffmpeg, Exiftool	User-based
Vegas Pro (280 videos)	Vegas Pro	User-based

are presented in separate paragraphs. An overview of the datasets involved is provided in Table 1.

1) AI-BASED MANIPULATION

We developed a dataset of videos manipulated with some recently-proposed video inpainting techniques. Such technologies allow removing arbitrary areas and objects from video frames, and have been chosen since they allow to work on scenes without people or faces. A set of 312 original videos have been collected from Youtube 8M [26], a video dataset with no copyright restrictions on Youtube, Sport 1M [27], Socrates [28], and VISION [29]. The original videos collected from the aforementioned datasets depict different scenes, from outdoor to urban environments, with resolutions ranging from 720p to 1080p. For each video, we semi-automatically generated the masks that identify the object to remove, and we generated different inpainted versions with a resolution of 432 × 240.

In our experiments, the following three technologies have been exploited:

- Spatial Temporal Transformer Network (*STTN*) [30];
- Onion Peel Network (*OPN*) [31];
- Generative Multi-column Convolutional Neural Networks (*GM-CNN*) [32].

The first two techniques are conceived for video inpainting and exploit both spatial and temporal information, while the third one is a powerful image inpainting technique that we apply frame-by-frame. Moreover, the data inpainted with *OPN* are post processed by the same inpainting technique using a Temporal Consistency Network (TCN) [33] aimed at removing temporal inconsistencies such as flickering in the inpainted area. These three pipelines have been applied to generate three different toolchains, using the implementation provided by the authors of the papers.[5] Examples of the original frames, the masks, and the resulting inpainted frames are reported in Figure 4.

[5]https://github.com/shepnerd/inpainting_gmcnn
https://github.com/seoungwugoh/opn-demo
https://github.com/researchmm/STTN

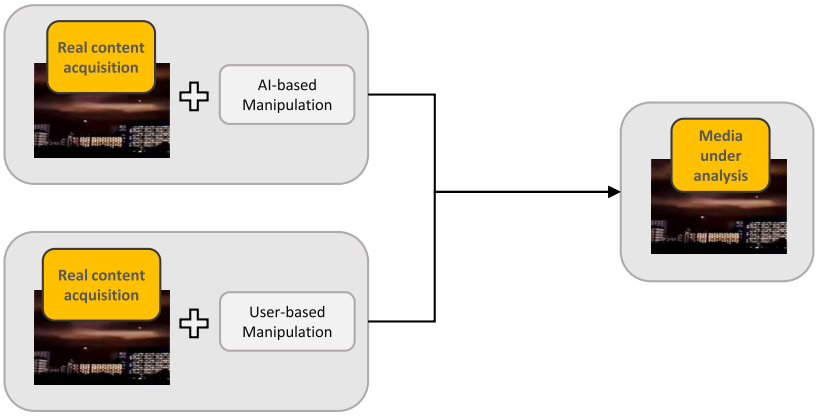


FIGURE 3. Functional blocks for toolchains considered for experimental setup 1.

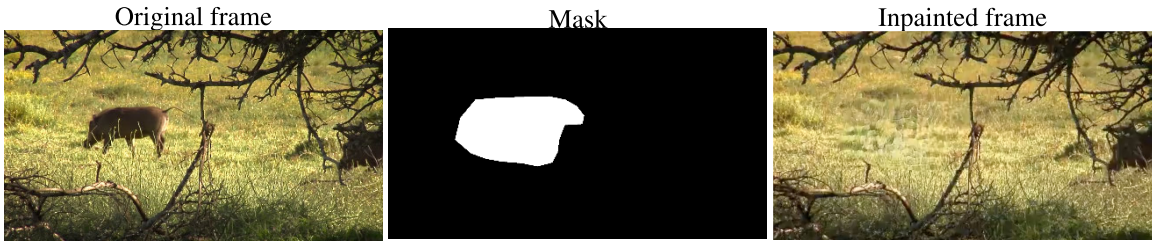


FIGURE 4. Example of inpainting process.

In total, we generated $312 \times 3 = 936$ manipulated videos, in addition to their pristine versions.

2) USER-BASED MANIPULATIONS

We considered the EVA-7K dataset [34], which consists of 6860 videos altered by software and exchanged through several social media platforms. This dataset provides *software-based manipulations* obtained from the same pristine contents, thus providing a good experimental basis for the analysis of different toolchains.

The 140 pristine videos are edited with the following software:

- *Adobe Premiere*[6]*:* each video was manually cut by keeping 5 to 7 seconds and saved as H.264 with medium bitrate setting;
- *Avidemux*[6]*:* each video was manually cut by keeping 5 to 7 seconds and saved as *copy* and *MP4 Muxer* settings;
- *Exiftool*[6]*:* each video was manually processed to change the date information within the metadata;
- *Kdenlive*[6]*:* each video was manually cut by keeping 5 to 7 seconds and saved with the *MP4 - the dominating format(H264/AAC)* setting;
- *ffmpeg*[6]*:* the software was used in an automated way to (i) trim the video to 5 seconds and re-encode it with H.264/AVC; (ii) trim the video to 5 seconds by copying the audio and video coding parameters to minimize the traces left by the operation; (iii) trim the video to 15 seconds and slow it down by $1/4\times$; (iv) speed the video up by $4\times$ through *ffmpeg*; (v) trim the video to 15 seconds and downscale it to the resolution of 320×240.
- *Vegas Pro v.16*[7]: we cut 140 native videos from *EVA-7K* with *Vegas Pro* editing. Overall we built 140 videos edited with H.264/AVC and 140 edited with H.265/HEVC. The manipulation with *Vegas Pro* has affected the resolution, the duration, the frame per second, the audio codec, and the video codec for each video. The video resolution was set to FullHD (1920×1080 pixels) at 25 fps. We considered video encoding and transcoding with H.264/AVC and H.265/HEVC. In addition, we used AAC as audio codec. Finally, the original video was cut randomly with at least 5 seconds of video content.

B. FEATURES

The analysis of videos is performed by exploiting features from different domains. We consider container-based features including metadata, coding parameters, and video container structure, and content-based features extracted from state-of-the-art detectors.

[6]Videos from EVA-7K [34].

[7]Videos from Vegas-Pro dataset are accessible at https://drive.google.com/drive/folders/1w5XYbfgV4n3n_c_xYu6v57R37ysOV6jd?usp=sharing.

1) CONTAINER-BASED FEATURES

The video container analysis is based on the recent techniques proposed in [34], [35]. The video file (or container) is represented as a labeled tree where internal nodes and leaves correspond to atoms and field-value attributes. A video container X can be characterized by the set of symbols $\{t_1, \dots t_m\}$, where t_i can be: a *field-symbols*, i.e. the path from the root to any field; a *value-symbols*, i.e. the path from the root to any field-value. An example of this representation can be[8]:

$$\begin{aligned} t_1 &= [\texttt{ftyp/@majorBrand}] \\ t_2 &= [\texttt{ftyp/@majorBrand/3gp4}] \\ &\dots \\ t_i &= [\texttt{moov/mvhd/@timescale}] \\ t_{i+1} &= [\texttt{moov/mvhd/@timescale/1000}] \\ &\dots \end{aligned}$$

Overall, given a set of possible origins $\mathcal{O} = \{O_1, \dots, O_t\}$ (e.g., AI-based inpainting, user-based manipulation), the method exploits Decision Trees, a non-parametric learning method, to assign a container X to a specific class O_u based on its symbols $\{t_1, \dots, t_m\}$. The method is enriched with a likelihood ratio framework designed to automatically clean up the container elements that only contribute to source intra-variability.

2) CONTENT-BASED FEATURES

Content-based analysis focuses in particular on the characterization of different AI-based inpanting manipulation techniques. We start the analysis by exploiting the methodology proposed in [36], which consists in applying a convolutional network without fully-connected layers, which has the advantage of accepting input media with arbitrary sizes, and returns a full resolution tampering probability map with values in [0, 1] for each video frame. The map is used as a raw data to evaluate the detection capabilities of differently trained networks. A pictorial representation of the technique is reported in Figure 5. One specificity of the architecture is the pre-filtering module, which is intended to act as a high-pass filter to enhance the tampering traces left in the signal.

We fine-tuned the pre-trained models separately on frames inpainted with the different inpaiting techniques (GMCNN, OPN, and STTN), and with the pristine frames. We will refer to detectors trained separately on each inpainting technique, as S_1, S_2, and S_3 where $1 \rightarrow$ GMCNN, $2 \rightarrow$ OPN, $3 \rightarrow$ STTN, each one trained to detect manipulations generated with one of the toolchains considered. We observed that, when testing such networks on data from the three different toolchains (indicated as X_1, X_2, and X_3), the inpainted areas are typically more accurately localized when S_i is tested on X_i for the same index i. Therefore, we explored the possibility of leveraging frame responses of S_1, S_2, and S_3 to extract indications on the inpainting toolchain used. Figure 6 reports an example of network's output maps when testing on data from different toolchains.

On this basis, we define a statistics R to be extracted from each map. In particular, by denoting as $S(X)$ the output map of a network S from a frame X, we split the pixels in two sets as follows:

$$\mathrm{M} \doteq \{\text{pixels in } S(X) \text{ that are } \geq 0.5\} \tag{5}$$

$$\mathrm{P} \doteq \{\text{pixels in } S(X) \text{ that are } < 0.5\} \tag{6}$$

By denoting as med_{M} and med_{P} the median values of M and P, respectively, the final statistics R is defined as:

$$R = med_{\mathrm{M}} - med_{\mathrm{P}}. \tag{7}$$

Accordingly, R is intended to quantify the separation between M and P on a specific frame, and represents the content-based feature used to identify the correct toolchain instance.

For every case depicted in Figure 6, we defined statistical models, using equation (7), to detect the inpainting techniques though a majority voting criteria considering the likelihood with respect to the testing data.

C. EXPERIMENTAL EVALUATION

In this section we assess the capability of the proposed method to cluster together signatures of videos produced by the same unknown processing toolchain. To this purpose, we consider two toolchain classes: AI-based manipulations and user-based manipulations, as described above. The first class includes 3 different processing chains (STTN, OPN, GM-CNN), while the second one includes 7 different processing toolchains (Adobe Premiere, Avidemux, Exiftool, Kdenlive, Ffmpeg, Vegas Pro AVC, Vegas Pro HEVC).

The initial feature space of size 20319, obtained by concatenating container- and content-based features, has been compressed using a single-layer media signature encoder (as described in Section III-A) into a 25-dimensional latent space. This size has been selected to demonstrate the ability of the proposed method to maintain strong discriminative power even when compressing the feature space into a minimal number of elements.

We performed a preliminary step to assess the impact of the signature encoding process on the features discrimination power. For this purpose, we considered two SVM classifiers, the first one built on the considered features while the second one built on the encoded signatures. In Figure 7 we report the accuracy in the form of a confusion matrix computed over the 11 different toolchains. We can notice a slight performance drop in a few categories due to the signature encoding, which is however limited to 6% on the discrimination power (average accuracy drop from 78.9% to 72.9%).

To evaluate open-world scenarios, we considered 10 cases in which each analyzed toolchain is not available in the training set. Therefore, we applied a leave-one-out strategy where we repeated the experiments 10 times, removing

[8]Note that @ is used to identify atom parameters.

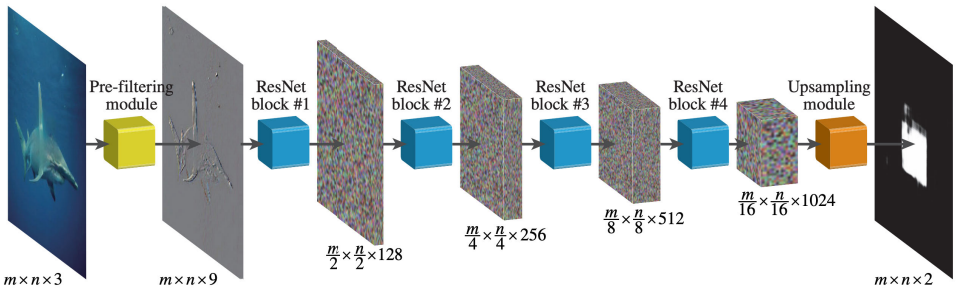


FIGURE 5. Architecture of the network used for inpaiting localization, as reported in [36].

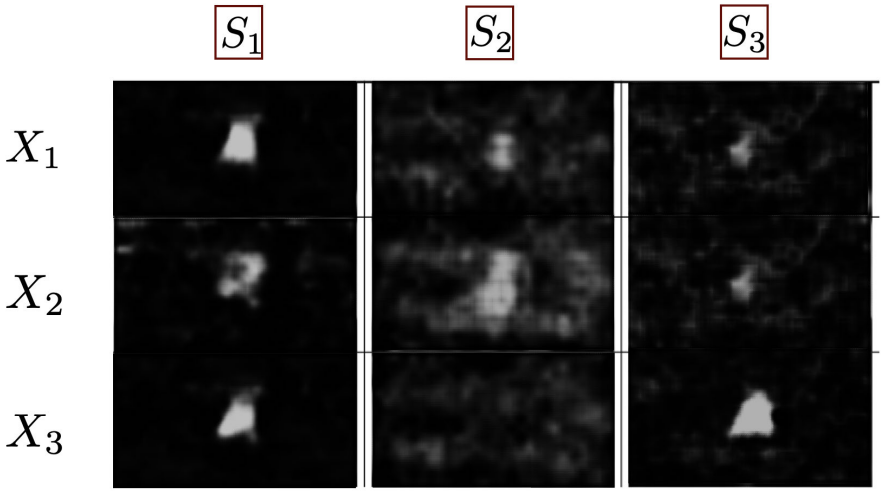
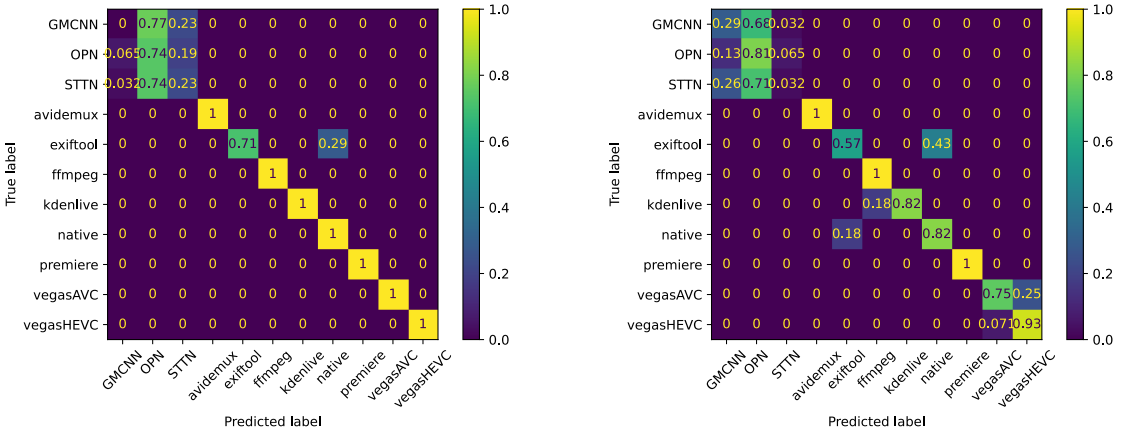


FIGURE 6. AI-based inpainting manipulation probability maps produced by different detectors (columns) on data coming from different toolchains (row).



(a) Classifier performance on considered features.

(b) Classifier performance on encoded signatures.

FIGURE 7. Assessment of the signature capability to encode the features information.

each time all the videos produced by one of the above toolchains. At test time we extracted the signatures from video items belonging to both known and unknown classes. Finally, we computed the Euclidean distances between pairs of signatures associated to unknown class (intra-distances), and between pairs of signatures associated to unknown and

known classes, respectively (inter-distances). Our aim is to show that unknown signatures generated from content processed by the same unknown toolchain are not only separable from those associated to other known toolchains (high inter-distances), but also close to each other in the signature space (low intra-distances).

We represent the obtained results graphically, reporting for each experiment the distribution of intra-distances for the unknown class, and the inter-distances between the unknown class and each of the known ones. In order to capture the most significant statistical aspects of each distribution, we represent them with a combination of box- and violin-plots. We report an example in Figure 8: each box represents the range between the first and the third quartiles, with an orange line in between representing the median; whiskers extend to 1.5 times the interquartile range, and small circles indicate outliers (i.e., every datum outside that interval). At the same time, the blue violin-shaped plot underneath the box represents a density estimation of data. Using this representation, we can easily assess whether the learned signatures can be used to identify the unknown class. Indeed, signatures are effective in discriminating a class when the bulk of intra-distances density is lower than inter-distances densities. Moreover, the relatively small range for intra-distances suggests that they form a cluster easily identifiable as a new class. In Figure 8, for instance, we have a distribution of intra-distances, related to *Avidemux*, which is below most of those of the inter-distances with other classes, except for *ffmpeg* and partly *Kdenlive*. We can thus infer that when *Avidemux* is unknown at training time, contents produced by it can be wrongly identified as produced by *ffmpeg* (as the overlap between distributions is large) or, more rarely, by *Kdenlive* (as the overlap is smaller).

Our experiments yielded 10 distinct outcomes, one for each potential unknown class. We provide all achieved results in Figure 9 and 10 for an overall understanding of the system performance. In the following we report and discuss the most significant cases.

1) CASE 1: FFMPEG-BASED INSTANCES

This case represents the analysis of an unknown software that implements parts of the *ffmpeg* library. In Figure 9 we show the intra-variability of *Avidemux* signatures and their inter-variability with respect to all available toolchains. It can be noticed that the proposed features can cluster unknown data as media belonging to similar toolchains (leftmost plot). Other available toolchains are generally far from *Avidemux* in the signature space, except for *ffmpeg*, which highlights a relevant similarity. This is interesting, since it shows that the proposed method allows both to cluster new data and to find similarities with related available toolchains. We observed identical results in our analysis of data processed through both the *Kdenlive* and *ffmpeg* toolchains.

2) CASE 2: AI-BASED INSTANCES

This case allows examining unknown instances from AI-based manipulations, i.e., OPN, STTN, or GMCNN. In Figure 10 we show the results for GMCNN (very similar results were obtained with OPN and STTN). As in the previous case, we found that the proposed signatures can cluster unknown toolchains. In this case, however, unknown signatures show a higher degree of compatibility with the other available AI-based toolchains (see Figure 10), making it hard to properly separate each instance. Nevertheless, the distribution of the achieved signatures is strongly separable from user-based manipulations. Therefore, we cannot expect to identify the specific AI-based toolchain, but we are able to find a high compatibility with toolchains of a similar pipeline (since the AI-based manipulations share similar pipelines).

3) CASE 3: INSTANCES OF AVAILABLE TOOLCHAINS WITH DIFFERENT SETTINGS

Within our reference dataset, we have *Vegas Pro* instances encoded with different settings. We tested each of the available settings as an unknown toolchain. We found that changing the setting of the encoding process marginally affects the signature. Indeed, in the encoded space, unknown signatures belonging to *Vegas Pro* HEVC are highly compatible with their AVC kindreds. We report the results for the case of *Vegas Pro* HEVC signatures in Figure 10. Conversely, the unknown toolchain forms a cluster that is highly separable from other available toolchains. We achieved very similar results for the AVC case.

4) CASE 4: MISCLASSIFIED INSTANCES

When dealing with unknown instances from Adobe Premiere, we noticed that they are still clustered together in the signature space (see Figure 9). However, the separability from *Vegas Pro* HEVC instances is not so sharp, possibly leading to misclassified instances. The Adobe Premiere tool is the only case in which this issue occurred.

D. MULTIPLE UNKNOWN INSTANCES

The abundance of available classes in this scenario allowed us to stress the system's capabilities by increasing the number of unknown toolchains. We performed a leave-two-out strategy: in each test we excluded two classes from the training. It is worth noticing that this setup generates 55 tests (the number of pairs within a sample of 11 classes), thus making it unfeasible to report the results in the form of violin plots. Then, we considered the following main distributions:

1) the intra-distribution of the first unknown class (C1-C1);
2) the intra-distribution of the second unknown class (C2-C2);
3) the inter-distribution between the two unknown class (C1-C2);
4) the inter-distribution between the first unknown class and the available classes (C1-All);
5) the inter-distribution between the second unknown class and the available classes (C2-All);

More specifically, intra-distances (C1-C1 and C2-C2) highlight the capability of the system to cluster the signature of

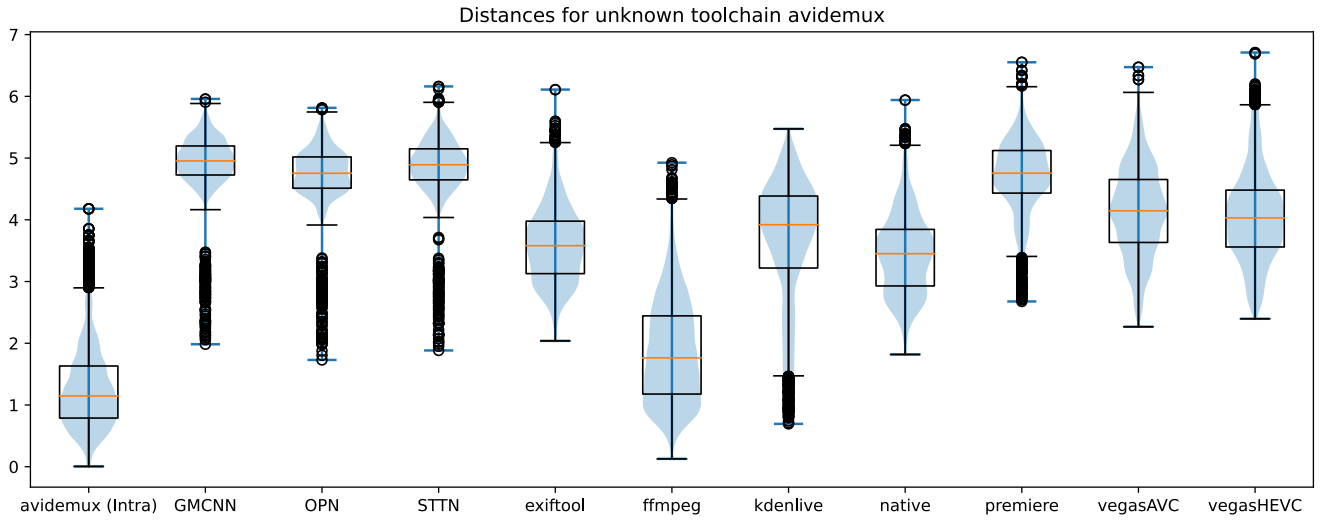


FIGURE 8. Avidemux signatures intra-distances (leftmost plot) and inter-distances related to all known toolchains instances.

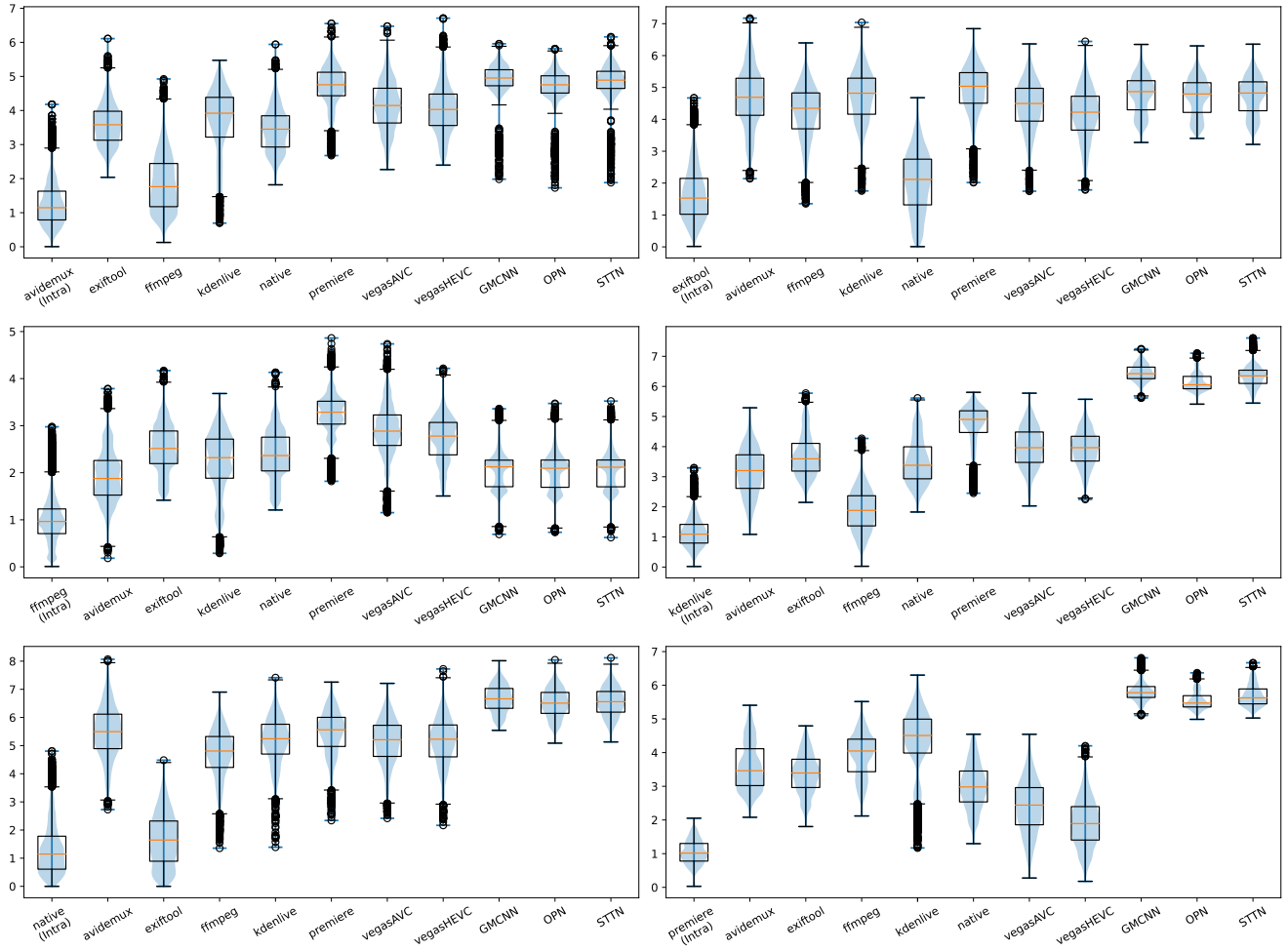


FIGURE 9. Signatures intra- and inter-distances distributions for available toolchains in leave-one-out strategy (part 1 of 2). The unknown toolchain intra-distribution is shown in the left-side of each plot. Inter-distances are reported with all available toolchains.

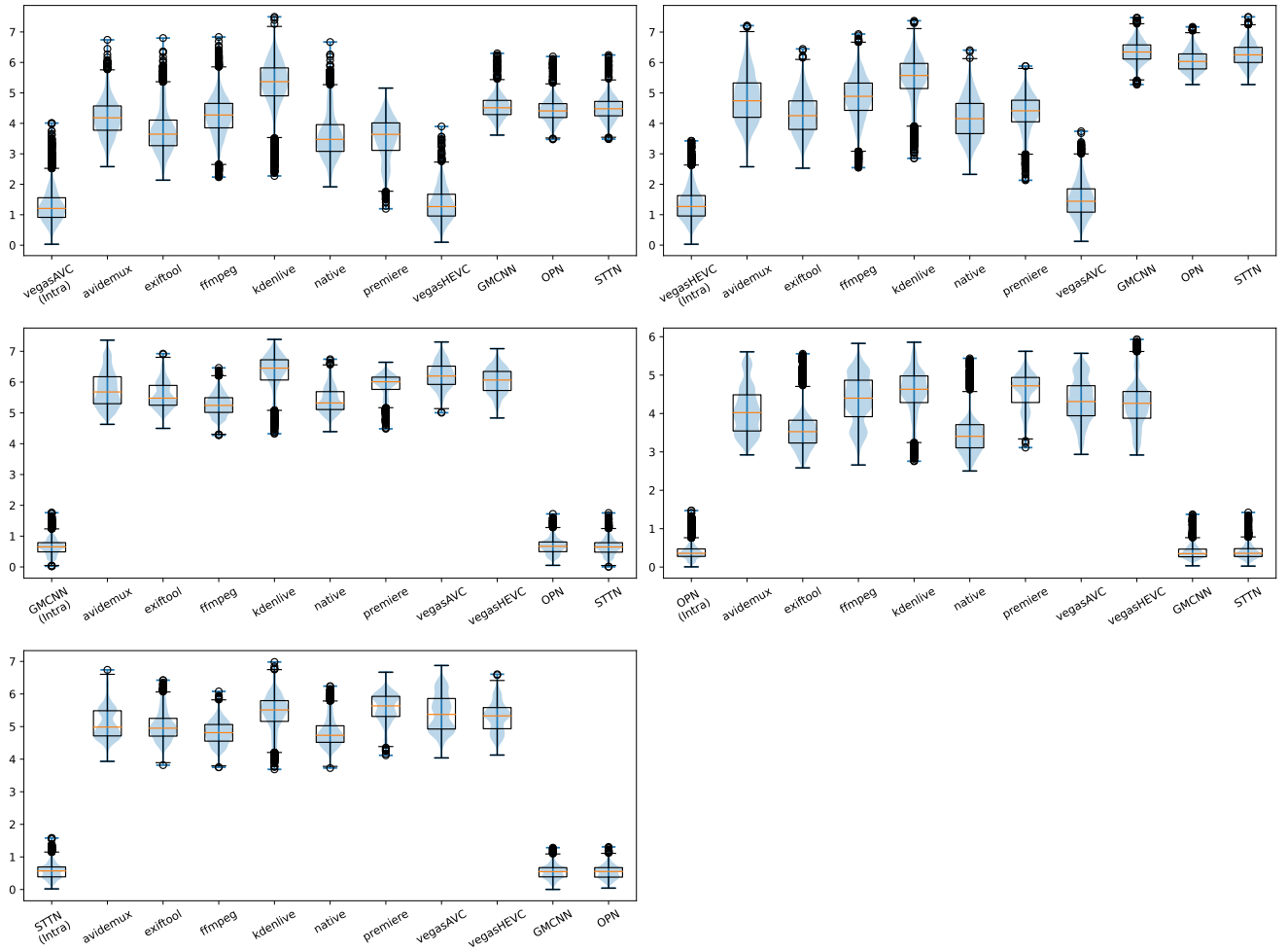


FIGURE 10. Signatures intra- and inter-distances distributions for available toolchains in leave-one-out strategy (part 2 of 2). The unknown toolchain intra-distribution is shown in the left-side of each plot. Inter-distances are reported with all available toolchains.

each unknown class. The inter-distance C1-C2 summarizes the capability of the system to identify the two unknown toolchains as different instances. Similarly, the inter-distances C1-All and C2-All, summarize the capability of the system to identify each unknown toolchain as something new with respect to the available signatures. In Figure 11 we report the above statistics for each experiments by plotting each distribution range between 10% and 90% percentiles. We also report the results of the Kolmogorov-Smirnov test [37], a nonparametric hypothesis test used to measure to which extent two underlying one-dimensional probability distributions differ. The test's output expresses the difference between the cumulative distribution functions of the empirical distributions of the two samples over the data range.

In most cases, the distributions C1-All and C2-All confirm the system trend to distinguish both unknown classes from the available toolchains. Similarly to the leave-one-out test, some overlapping are found when related toolchains are excluded (e.g. *ffmpeg* vs *Kdenlive*). This is reasonable since the system identifies some similarities that actually exist among toolchains. The only relevant error is found when *Vegas Pro* and *Premiere* are excluded. Even if they are still distinguished from the available toolchain, they expose a strong degree of overlapping between them. This is not surprising since they produced some errors even in the leave-one-out experiment.

E. UNKNOWN AI-BASED FAMILY

To evaluate the generalization capabilities of the framework, we removed all AI-based instances (OPN, STTN, and GMCNN) from the training. In this case, we report the results in the form of a confusion matrix containing the average Euclidean distances among each instances pair (see Figure 12). It can be noticed that the average intra-variability of the AI-based family is lower than any other inter-variability in the matrix. This suggests that even when the AI-based family is completely unknown, the extracted signatures cluster together in the encoded space. At the same time, the separability among the three instances looks harder. We must consider, however, that the discrimination capability of the proposed features is still not particularly accurate for AI-based instances even in a fully informed scenario.

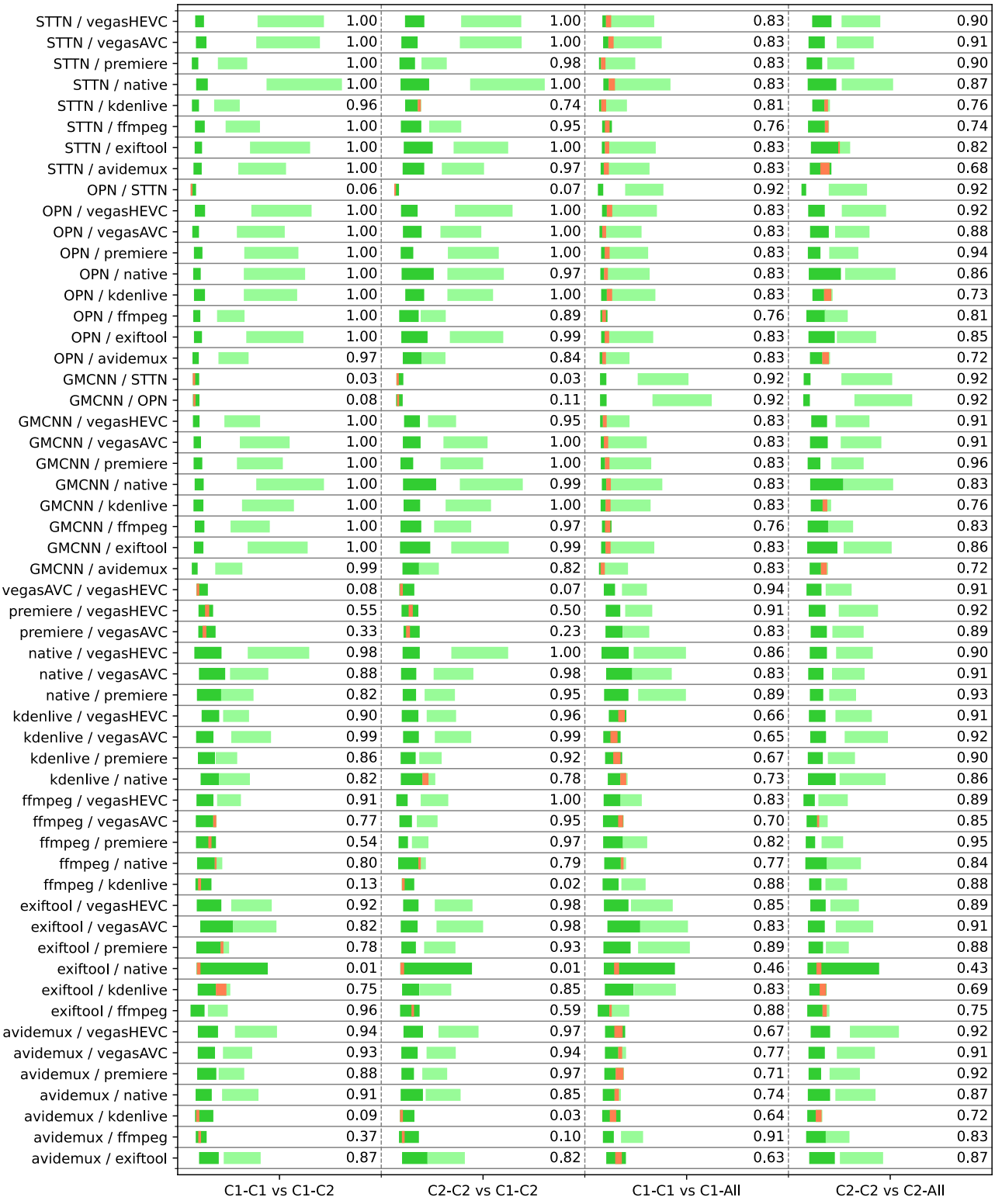


FIGURE 11. Intra- and inter-distances distribution when two classes are left out of the training set. The range between 10% and 90% of each distribution is reported in two shades of green. The red portion highlights the overlapping degree between the two distributions. We report for each experiment the result of the two-sample Kolmogorov-Smirnov test.

V. EXPERIMENTAL SETUP 2: MEDIA4COMMUNITY

In **media4community** setup, we focus on the realistic case where the provider does not apply any manipulation detection before uploading the media, thus possibly spreading deceptive information to the user community. In this context, we aim at giving the possibility to retrieve the lifecyle

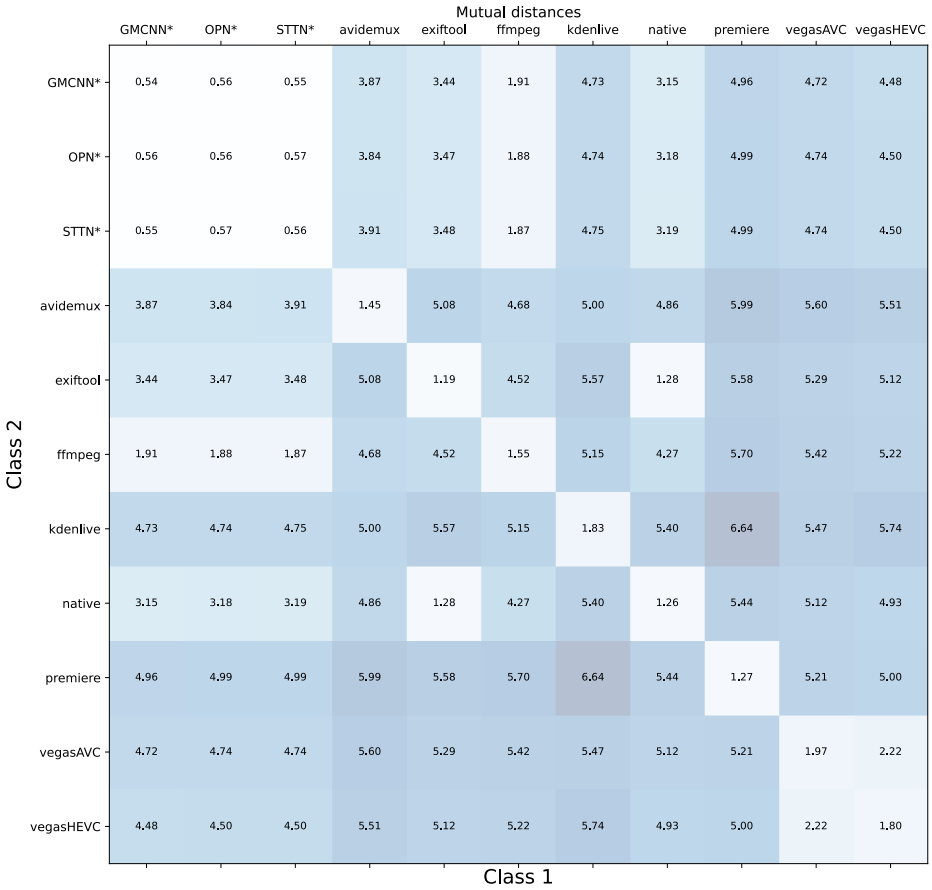


FIGURE 12. Average intra and inter-distances among instances in the case of unknown AI-based instances.

of a media in terms of sharing history, by focusing on the characterization of images that have been recycled (shared multiple times) from one social network to another. We consider toolchains including native media, shared media (media that have been natively produced/acquired and then shared), and recycled media (media that have been downloaded from another service and re-shared). The considered scenario is depicted in Figure 13. Additionally, for this configuration, we present the datasets utilized, the features employed, and the results obtained on open-world data in the subsections below. A list of the involved datasets is summarized in Table 2 where, with *Facebook, Instagram, Telegram, Twitter and WhatsApp* we refer to images of the FODB dataset [38] shared on these social networks and, with *FB, TW and FL* to images shared on Facebook, Twitter and Flicker of the R-SMUD dataset [39].

A. DATASETS

We employed datasets of digital images that have been shared through different platforms once or multiple times, thus yielding various toolchains. This has been done through semi-automated procedures for uploading and downloading contents to and from different platforms. According to the definition of the **media4community** experimental setup, we collected natively-shared media, i.e., native data shared on a social network only once, and recycled media, i.e., data shared twice or more through the same or a different platform. We indicate natively-shared media and recycled media as follow:

$$[P] = \{\text{native data uploaded to platform } P \text{ then downloaded}\}$$
$$[P' \rightarrow P] = \{\text{data in } P' \text{ uploaded to } P \text{ then downloaded}\}.$$

TABLE 2. Summary of the considered datasets for the Experimental Setup 2.

Dataset -	Manipulation Instance	Manipulation Type
FODB (23106 images)	Facebook, Instagram, Telegram, Twitter, Whatsapp	Natively Shared
R-SMUD (35100 images)	FB, TW, FL	Natively Shared Recycled Images

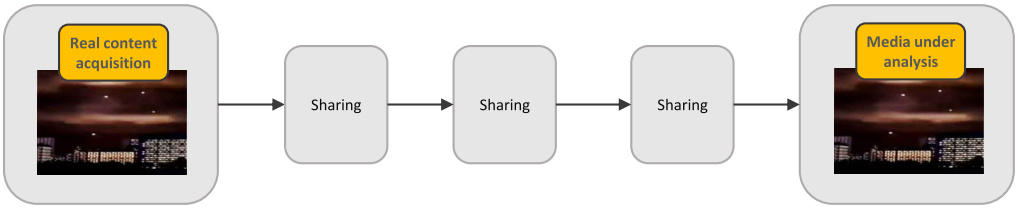


FIGURE 13. Functional blocks for toolchains considered for experimental setup 2.

We used two image collections featuring multiple shares, namely *R-SMUD* [40] and *FODB* [38], containing both natively-shared and recycled media:

- *R-SMUD*: 50 RAW images are extracted from the RAISE dataset [39]. Then, the images are top-left cropped with three different sizes (377 × 600, 1012 × 1800, and 1687 × 3000) while keeping an aspect ratio of 9:16. Moreover, all the different cropped images are compressed using The Independent JPEG Group's JPEG software under six quality factors (50, 60, 70, 80, 90, 100) before being uploaded. All images are shared up to three times on the following platforms: Facebook (FB), Flickr (FL), Twitter (TW).
- *FODB*: 143 scenes acquired by 27 different smartphones for a total of 3651 images [38]. Each image is shared on five different platforms: Facebook, Instagram, Telegram, Twitter, WhatsApp. Uploads are done through the mobile apps installed on the respective devices. The database thus contains a total of 23106 JPEG images.

In this scenario, examples of toolchain instances are: 'One-time sharing with Facebook', 'Re-sharing from Twitter to Facebook'.

B. FEATURES

In order to instantiate the framework described in Section III, we considered the feature representation proposed in [41], which includes both content-based (*DCT*) and container-based features (*META* and *HEADER*).

In particular, the following feature vectors are used:

- *DCT*: histograms of the DCT coefficients (9 AC subbands) are computed from the full image and concatenated; integer values between −20 and 20 have been considered (41 bins) in each subband, for a total feature size of 369;
- *META*: metadata related to the JPEG compression settings of the image under investigation; the 152-dimensional feature vector encodes information on the quantization tables of luminance and chrominance channels, the Huffman encoding tables, optimized coding options and progressive modes, the image size;
- *HEADER*: this 8-dimensional feature is defined starting from the analysis of the JPEG header of the file under investigation; this is a novel approach that scans the structure of the image container and counts the frequency of 8 selected types of segments found in the file header.

C. EXPERIMENTAL EVALUATION

Within this experimental configuration, we examined three distinct categories of processing toolchains, specifically:

1) **NS**, containing single-shared images;
2) **R1**, containing recycled images, i.e., sharing chains of length equal to 2;
3) **R2**, containing twice recycled images, i.e., sharing chains of length equal to 3.

As seen in the **media4provider** scenario, the considered 529 features have been fused using a media signature encoder as described in Section III-A. Here, we also utilized a small 25-dimensional latent space produced by a single layer media signature encoder to showcase the performance of the proposed method even when the feature space is minimized to just a few elements.

In following Section IV-C, we first evaluated the proposed method's accuracy for the **media4community** scenario in a closed-world environment, prior to its application in an open-world context. The accuracy results are displayed as a confusion matrix in Figure 15.

Although a slight performance drop is noticeable for a few categories here as well, the signature encoding produces a limited drop of 6% on the discrimination power (average accuracy drop from 42.6% to 36.9%).

D. OPEN WORLD EXPERIMENTS

We consider two cases in which the analyzed toolchain is not available in the training data:

- **Unknown time-instances:** the considered social media is available in the training set, but the training data belong to past years, thus possibly exhibiting different coding artifacts or metadata features, as the uploading algorithms get updated over time.
- **Unknown toolchains:** the social media corresponding to the query image are available in the training data but it has been subjected to multiple exchanges not available in the training data.

Results are reported in the following paragraphs.

1) CASE 1: UNKNOWN TIME-INSTANCES

To assess the behavior of our signatures when different *time-instances* of a given sharing platform are present in

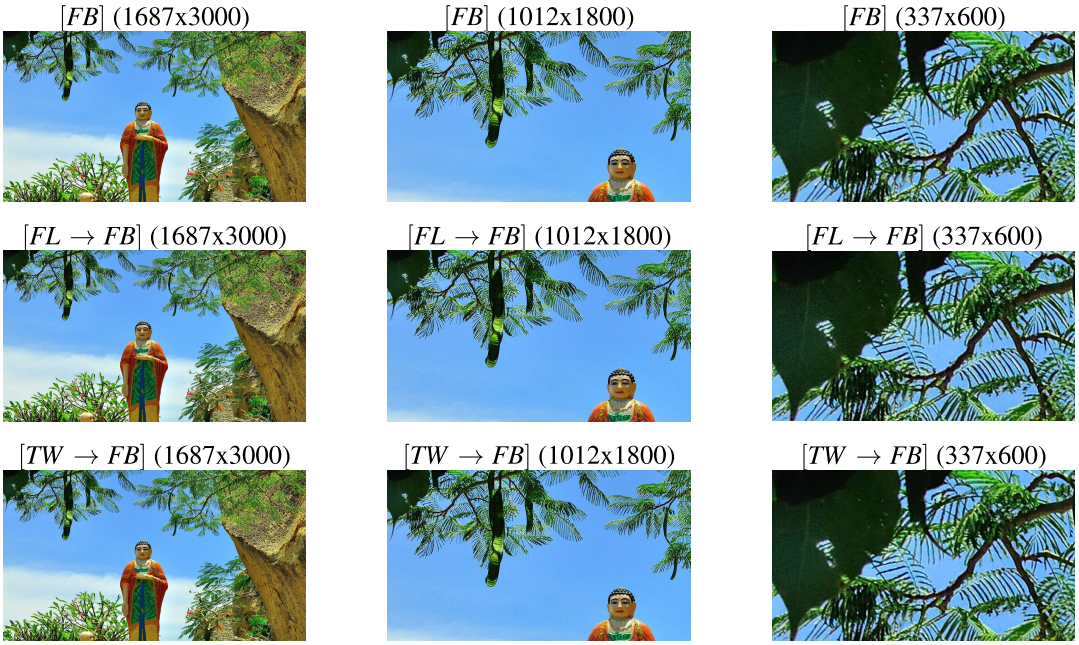
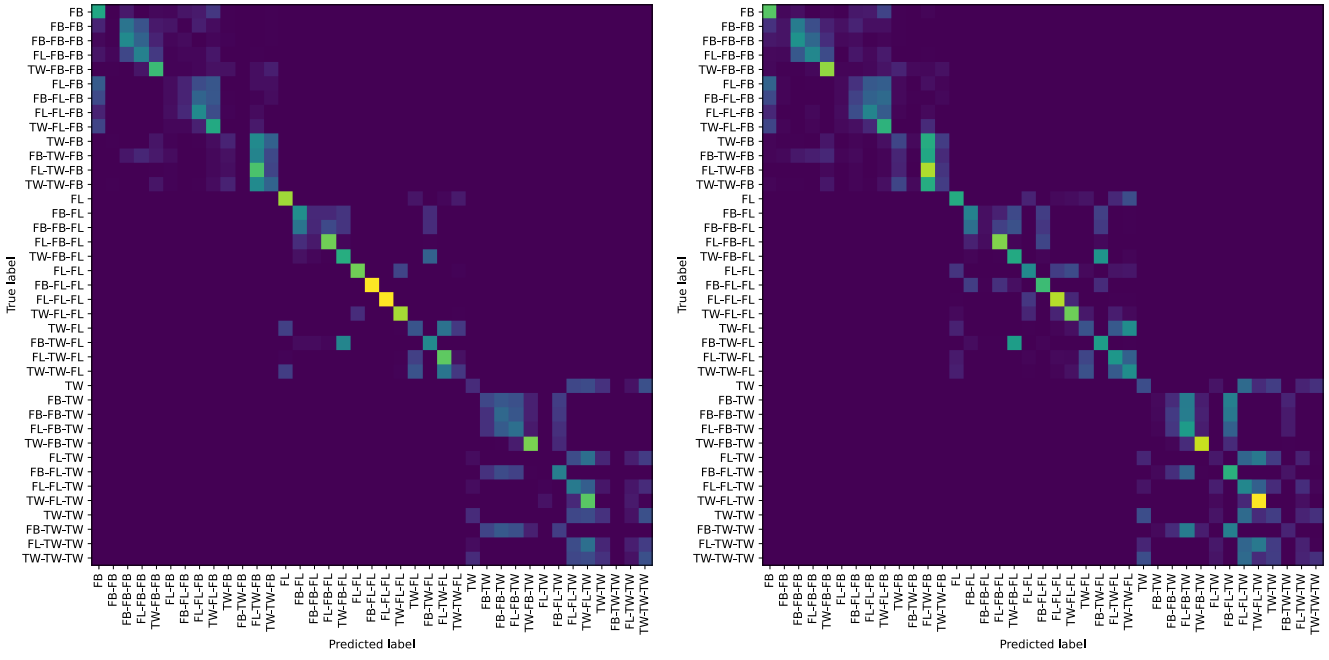


FIGURE 14. Example images from *R-SMUD*.



(a) Classifier performance on considered features.

(b) Classifier performance on encoded signatures.

FIGURE 15. Assessment of the signature capability to encode the features information in m4c scenario. Results are reported in the form of confusion matrix. Warmer colors are associated to higher detection accuracies. Labels are arranged based on the last social media exchange. We notice that the detection of the last processing step is generally achieved while the system can fail in determining previous sharing. Overall, we achieve an accuracy of 42.6% and 36.9% before and after the signature encoding respectively.

the test set, we made use of both R-SMUD and FODB datasets combined. R-SMUD, collected in 2017, contains three type of processing chains (NS, R1, R2) related to three platforms (*Facebook*, *Flickr*, *Twitter*). FODB, collected in 2020, contains a single toolchain (NS) related to five platforms (*Facebook*, *Instagram*, *Telegram*, *Twitter*, *WhatsApp*). We used 80% of R-SMUD (NS only) as training set and the whole FODB, together with the remaining 20% of R-SMUD, as test set. We trained the signature-generation autoencoder on the training set and then extracted the

signatures from all samples in the test set. Finally, we compared the intra-class distances with the inter-class distances for each tested class.

Figure 16 reports an ensemble of results obtained on the described cross-dataset scenario. Each plot refers to a specific class (highlighted by a "*" character) displaying the distribution of intra-distances for that class and the distributions of inter-distances from that class to all the others. R-SMUD's classes are denoted by the shortened versions of the platforms' names (*FB*, *FL*, *TW*), while FODB's ones are reported with their full name and in parentheses, to denote that those were not present in the training set.

Interestingly, we can observe that the most evident overlaps happen for different time-instances of the same platform. In the top-left plot, R-SMUD's *Facebook* overlaps with FODB's *Facebook* (and partially with *WhatsApp*). Similarly, in the bottom-left plot, R-SMUD's *Twitter* only overlaps with FODB's *Twitter*. This is a key result in the evaluation of our system, as it suggests that the proposed signatures is able to recognize a known platform even when the image under analysis was shared at a different time from that of the training of the system. This clustering can be better visualized in Figure 17, which displays the average distances among instances.

Additionally, in the bottom-right plot we can further appreciate the partial overlapping of *WhatsApp* with both time-instances of *Facebook*, which is reasonable as the two are commercially related and are likely to share some common software features. Finally, in the top-right plot, we can observe a partial overlapping of *Flickr* and *Telegram*, possibly deriving from the lower level of compression introduced by these two platforms compared with the others. Also, note that we kept *Instagram* out of these results as it produces very different signatures from all other platforms, and thus the distributions of distances went off the scale. The very good identification of this class can also be appreciated in Figure 17.

2) CASE 2: UNKNOWN PROCESSING TOOLCHAIN

To simulate the presence of an unknown toolchain, we adopted a leave-k-out strategy, by excluding the whole R2 family (triple sharing chains) from the R-SMUD dataset. Thus, our training set was composed of single and double sharing chains, while triple chains only appeared in the test set.

As in the previous cases, the trained system is not aware of the existence of R2 chains and the experiment cannot be considered a classification problem. In this specific case, however, the best outcome we can expect is the following: given an unknown chain R2 in the form P_3P_2-P_1 (with P_i being a generic sharing platform), the framework produces a signature in the cluster of *P2-P1*, i.e., the known chain in R1 that coincides with its trailing part.

We trained the signature-generation autoencoder with the features extracted from NS and R1 families. Then, we deployed the trained encoder to calculate the signatures for all samples in R2. With the obtained signatures, given a specific chain in R2, we computed the Euclidean distances among samples of that chain (intra-distances) and from that chain to all known ones in NS and R1 (inter-distances), and we repeated this process for each of the 27 chains in R2.

Figure 18 reports the distance distributions obtained for six example chains in R2. In each plot, the leftmost distribution corresponds to the intra-distances for the current unknown chain (label in parentheses), while all the others are inter-distances with respect to each chain in NS and R1.

First, we can observe a general trend across all the reported cases: the lowest inter-distances are associated to all the known chains that share the last platform with the unknown chain. For instance, in the top-left plot, related to the chain *FB-FL-FB*, the lowest inter-distances are found for *FB*, *FB-FB*, *FL-FB* and *TW-FB*. Furthermore, in the two cases in the middle row, we observe that, among these four closer chains, the closest one is precisely the training part of the unknown chain under analysis (*FL-FL* for *FL-FL-FL* and *FB-FL* for *TW-FB-FL*). Similarly, in the two bottom cases, if we take a look at the four chains closest to the unknown one (*FB-TW*, *FL-TW*, *TW*, *TW-TW*), we can note that the chain with the highest distances is the one containing a platform that does not belong to the unknown chain (*FB-TW* for both *FL-TW-TW* and *TW-FL-TW*).

VI. DISCUSSION OF RESULTS

In the previous sections we demonstrated the versatility of the proposed approach by applying it to two separate open-world experimental setups, where we have instantiated data and features through the designed framework for the forensic analysis of digital images and videos. The experiments performed on different datasets highlighted the following results in the two considered experiments.

In all considered tests, we found that unknown toolchains can be identified as a separated cluster in the signature space, thus allowing forensic analysts to examine media that have been subjected to novel or unknown life cycles In our opinion, this is the most relevant contribution of the proposed work since it opens the path to real-world forensic media analysis.

In particular, in the **media4provider** scenario, *Vegas Pro* toolchains are also found to be similar independently of the coding parameters, thus highlighting a potential robustness of the signature to slight variations of the encoding software parameters. We also found that, when the unknown toolchain partially shares the life cycle of some available toolchains, a non-marginal degree of compatibility can be found. This is the case, for instance, of ffmpeg-based instances and AI-based manipulations. In a single case (*Vegas Pro*) we found that a non-marginal compatibility can be found with a different cluster.

On the other hand, in the **media4community** scenario, we are able to identify recycled images across multiple social

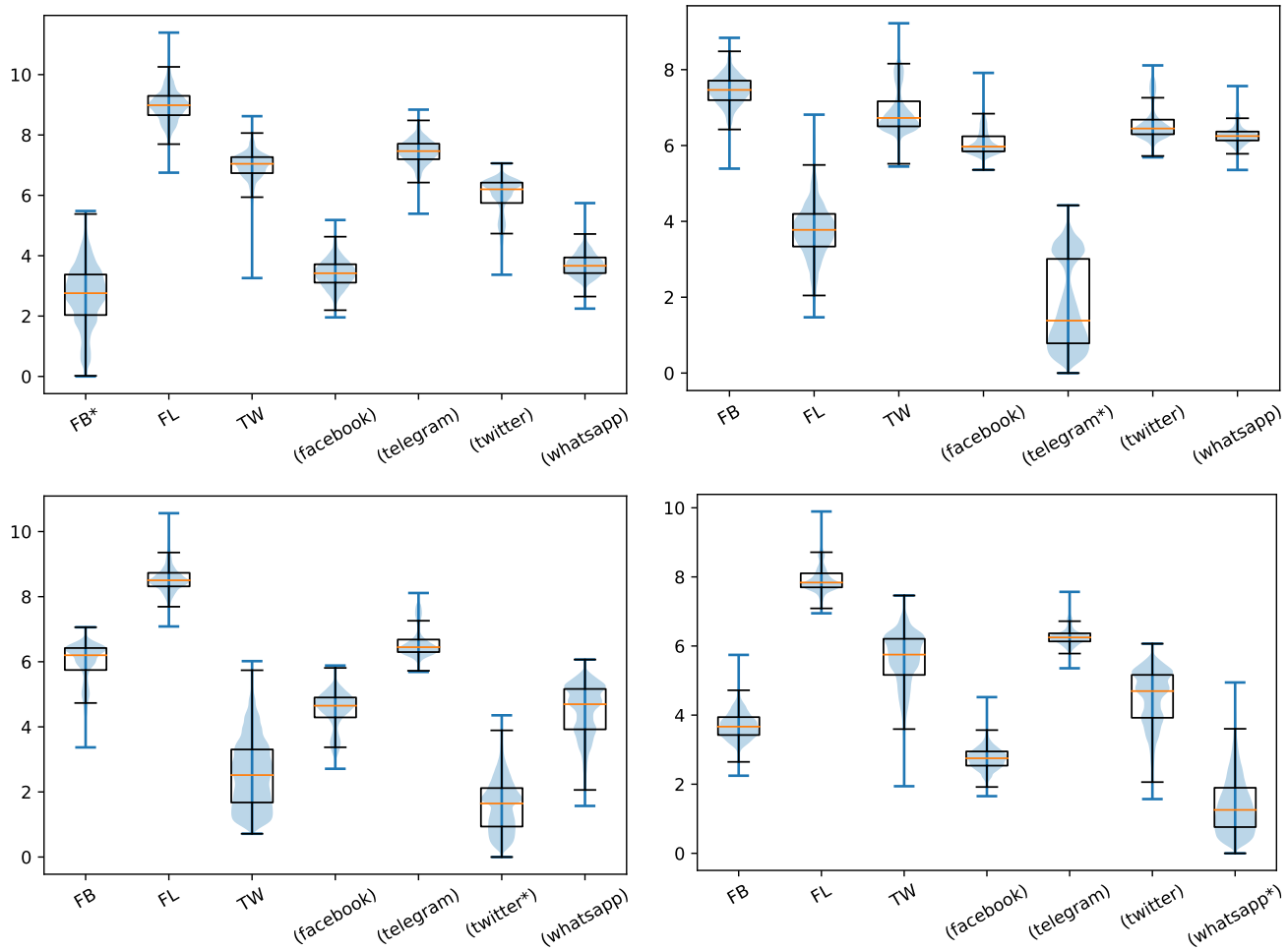


FIGURE 16. Distributions of intra-distances (labelled with '*') and inter-distances in a cross-dataset scenario, where R-SMUD (classes *FB*, *FL*, *TW*) is used as training set and FODB (classes *facebook*, *telegram*, *twitter*, *whatsapp*) is added in test.

networks. Furthermore, when a more complex sharing chain is analyzed, we are able to identify the most similar toolchain available in the training set (e.g., *TW-FB-FL* can be identified as *FB-FL* if *Twitter* is not known).

Eventually, the designed media signature proved to be robust to social media encoding variations and processing updates in time. Indeed, the framework trained with *Facebook* and *Twitter* data acquired in 2017 can be exploited to characterize *Facebook* and *Twitter* images exchanged in 2020, since the corresponding media signatures highlight strong similarities in the encoded space.

Furthermore, in both experimental setups, we proved that even considering both container- and content-based features, the proposed media signature encoding allows characterizing the media life cycle in low-dimensional spaces. This aspect is particularly relevant to guarantee the framework's applicability to large volumes of data.

A. COMPUTATIONAL COMPLEXITY AND SYSTEM SCALABILITY

The challenge of scalability is particularly relevant in open-world contexts, where thousands or even more instances of media may require signature extraction. The computational complexity of or system can be characterized by two factors: (i) the cost of extracting features and generating signatures from new instances; (ii) the cost of retrieving and cross-indexing a growing amount of signatures.

The first cost, related to the signature generation, does not really affect the scaling capability since it is a one-time cost to project each content in the signature space. Moreover, in the proposed experimental setups, costs related to the extraction of image and video container features, image content features, and signature encoding are limited to few seconds per media[9] (see Table 3). The extraction of AI-based features is by far the most resource-intensive task in this process, requiring approximately 0.5 seconds per frame.

It's important to note that the cost-independent nature of feature extraction from media content allows for highly efficient batch processing of large datasets and long videos.

The second cost associated with signature retrieval and cross-indexing becomes increasingly significant as the dataset size grows in the signature space, making it a critical factor for scalability.

[9]computed on an Intel(R) Core(TM) i9-7940X CPU @ 3.10GHz

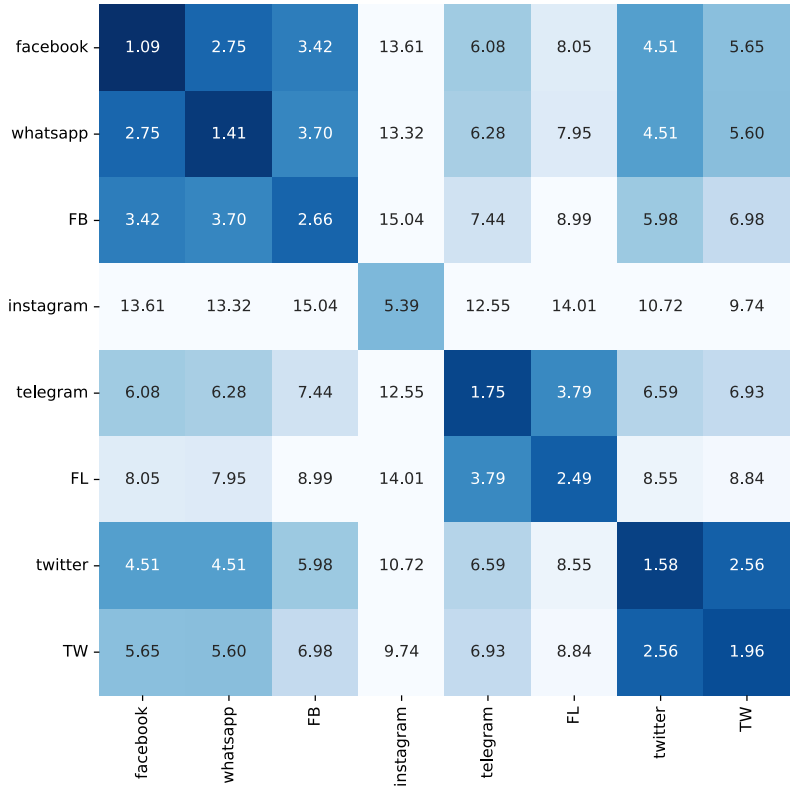


FIGURE 17. Average intra- and inter-distances in the R-SMUD/FODB cross-dataset scenario. R-SMUD's classes are *FB*, *FL*, *TW*; FODB's are *facebook*, *instagram*, *telegram*, *twitter*, *whatsapp*. Therefore, {*facebook*, *FB*} and {*twitter*, *TW*} are different time-instances of the same platform.

TABLE 3. Computational time required to extract features and compute signatures.

Operation	Cost
Video container-based features	$<$ 1 second per video
Video Inpaint features	$<$ 0.5 seconds per frame
Image header and DCT statistics	$<$ 10 seconds per image
Signature encoding	$<$ 1 second per media

TABLE 4. Indexing times (in secs) when using Milvus. The first three columns show results for different portions of a real signatures dataset containing 6480 items. The last two columns show results for a combination of real and synthetic signatures.

Index type	10%	50%	100%	$100\% + 10^4$	$100\% + 10^6$
IVF_FLAT	0.008	0.006	4.455	4.351	4.387
IVF_SQ8	0.014	0.007	4.901	4.941	4.914
IVF_PQ	0.005	0.009	0.004	0.004	0.050
RNSG	0.006	0.009	4.047	4.529	644.118
HNSW	0.008	0.011	4.835	4.368	19.149
ANNOY	0.010	0.008	4.020	4.086	14.196
FLAT	0.006	0.004	0.005	0.007	0.008

To quantify the influence of this cost on system scaling, we utilized Milvus,[10] a popular open-source search engine known for its ability to handle massive-scale feature vector indexing and retrieval tasks. Milvus comes with several options for building the index and to cross-index the signatures. In our test settings, including 6400 media, Milvus can build the index in less than 5 seconds and retrieve the nearest signatures with an average time lower than 0.1 seconds[11] when using the default indexing setting (IVF FLAT). However, other settings can be set to reduce the retrieval time at the price of longer time for building the index. In Table 4 we report the times required to build the indexes depending on the size of the signature database and the Milvus indexing setting.

Results for experiments on 10% and 50% of the dataset shows how, for small numbers of signatures, the time required to build the index is dominated by the system overhead, and thus the results are not meaningful. When building larger indexes we have an almost-constant cost for quantization-based indexes, while the time requirement of graph- and tree-based indexes grows rapidly with the number of items. It should be noted, however, that the index creation is rarely performed.

[10] https://milvus.io/
[11] computed on an Intel(R) Core(TM) i9-7940X CPU @ 3.10GHz

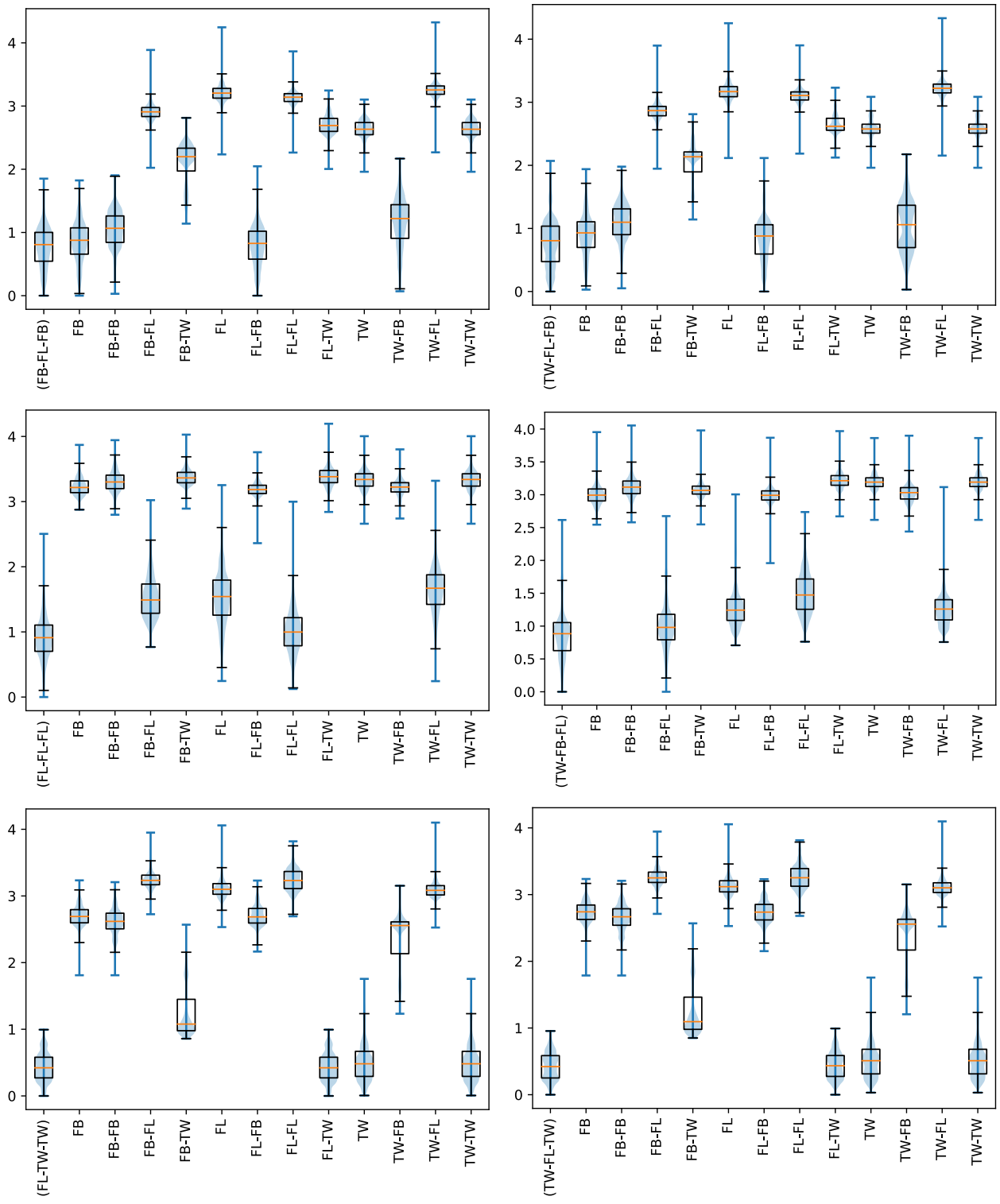


FIGURE 18. Distributions of intra-distances of a toolchain from the unknown R2 (leftmost plot) and inter-distances with respect to all known toolchains in NS and R1 (6 example cases out of the 27 toolchains in R2).

Similarly, in Table 5 we report the times required to retrieve similar vectors given a set of queries. In all cases we can retrieve the results almost in real-time. In our opinion Milvus, with its options IVF_FLAT and HNSW, strike the best balances between indexing and retrieval times. Moreover, both of them privilege a high recall rate

TABLE 5. Retrieval times (in secs) for our dataset. The first three columns show results for different portions of a real signatures dataset containing 6480 items. The last two columns show results for a combination of real and synthetic signatures.

Index type	10%	50%	100%	100% + 10^4	100% + 10^6
IVF_FLAT	0.054	0.023	0.072	0.070	0.987
IVF_SQ8	0.032	0.023	0.047	0.054	1.483
IVF_PQ	0.017	0.053	0.031	0.050	2.185
RNSG	0.017	0.032	0.031	0.028	0.413
HNSW	0.017	0.032	0.057	0.026	0.333
ANNOY	0.017	0.032	0.062	0.071	2.541
FLAT	0.017	0.030	0.031	0.054	2.044

over memory and time requirements. As we will use the indexing engine to retrieve similar toolchains, a high recall rate is an essential requirement of the tool that we will use.

VII. CONCLUSION

In this paper we introduced a framework for the forensic analysis of multimedia in open-world settings. We exploited a siamese architecture based on denoising autoencoders to encode multiple forensic features from different domains (content- and container-based features) into a compact descriptor. The proposed method is designed to cluster media belonging to similar toolchains in the signature space. We demonstrated the effectiveness of the proposed method by analysing two meaningful experimental setups involving both digital images and videos. Experimental results highlighted that the method is capable of clustering correctly media belonging to unfamiliar processing toolchains, thus allowing the identification of new and previously unknown life cycles. We also found that, when the unknown toolchain partially share the life cycle with one or more available toolchains, a non-marginal degree of compatibility is maintained in the encoded space, thus providing clues on the relevant life cycle. Finally, the proposed method has the potential to scale to internet volumes of information, given its capability to encode features in a low-dimensional space with limited computational effort.

This work can be considered a first step towards the design of a bigger picture for the investigation of media in open-world settings. Similarly to former fusion frameworks, the suggested method's primary drawback is that it is mostly dependent on the features that are fed into the network. In fact, different features might be more or less relevant in capturing traces left by new possible tampering operations, and their initial choice may have an impact on the overall capability of the system. Future research ought to focus on assessing the framework's robustness in terms of feature selection in characterizing unseen manipulations. Additionally, exploring the potential of incorporating new media types like digital audio and studying novel mathematical tools for signature generation can further enhance the framework's effectiveness.

ACKNOWLEDGMENT

The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

REFERENCES

[1] J. Hendrix and D. Morozoff, *Media Forensics in the Age of Disinformation*, H. T. Sencar, L. Verdoliva, and N. Memon, Eds. Singapore: Springer, 2022.

[2] J. Thies, M. Zollhöfer, M. Stamminger, C. Theobalt, and M. Nießner, "Face2Face: Real-time face capture and reenactment of RGB videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2387–2395.

[3] M. Ngo, S. Karaoglu, and T. Gevers, "Self-supervised face image manipulation by conditioning GAN on face decomposition," *IEEE Trans. Multimedia*, vol. 24, pp. 377–385, 2022.

[4] F. Lago, C. Pasquini, R. Bohme, H. Dumont, V. Goffaux, and G. Boato, "More real than real: A study on human visual perception of synthetic faces [applications corner]," *IEEE Signal Process. Mag.*, vol. 39, no. 1, pp. 109–116, Jan. 2022.

[5] S. J. Nightingale and H. Farid, "AI-synthesized faces are indistinguishable from real faces and more trustworthy," *Proc. Nat. Acad. Sci. USA*, vol. 119, no. 8, Feb. 2022, Art. no. e2120481119.

[6] L. Verdoliva, "Media forensics and DeepFakes: An overview," *IEEE J. Sel. Topics Signal Process.*, vol. 14, no. 5, pp. 910–932, Aug. 2020.

[7] S. Ganguly, A. Ganguly, S. Mohiuddin, S. Malakar, and R. Sarkar, "ViXNet: Vision transformer with xception network for deepfakes based video and image forgery detection," *Expert Syst. Appl.*, vol. 210, Dec. 2022, Art. no. 118423.

[8] K. Sharma, G. Singh, and P. Goyal, "IPDCN2: Improvised patch-based deep CNN for facial retouching detection," *Expert Syst. Appl.*, vol. 211, Jan. 2023, Art. no. 118612.

[9] J. Luka, J. Fridrich, and M. Goljan, "Digital camera identification from sensor pattern noise," *IEEE Trans. Inf. Forensics Security*, vol. 1, no. 2, pp. 205–214, Jun. 2006.

[10] M. Darvish Morshedi Hosseini and M. Goljan, "Camera identification from HDR images," in *Proc. ACM Workshop Inf. Hiding Multimedia Secur.*, Jul. 2019, pp. 69–76.

[11] S. Mandelli, P. Bestagini, L. Verdoliva, and S. Tubaro, "Facing device attribution problem for stabilized video sequences," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 14–27, 2020.

[12] S. Mandelli, F. Argenti, P. Bestagini, M. Iuliani, A. Piva, and S. Tubaro, "A modified Fourier-Mellin approach for source device identification on stabilized videos," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2020, pp. 1266–1270.

[13] A. Montibeller, C. Pasquini, G. Boato, S. Dell'Anna, and F. Pérez-González, "Gpu-accelerated sift-aided source identification of stabilized videos," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2022, pp. 2616–2620.

[14] M. Iuliani, M. Fontani, and A. Piva, "A leak in PRNU based source identification–questioning fingerprint uniqueness," *IEEE Access*, vol. 9, pp. 52455–52463, 2021.

[15] M. Iuliani, M. Fontani, D. Shullani, and A. Piva, "Hybrid reference-based video source identification," *Sensors*, vol. 19, no. 3, p. 649, Feb. 2019.

[16] T. D. Nguyen, S. Fang, and M. C. Stamm, "VideoFACT: Detecting video forgeries using attention, scene context, and forensic traces," 2022, *arXiv:2211.15775*.

[17] P. Zhou, N. Yu, Z. Wu, L. S. Davis, A. Shrivastava, and S.-N. Lim, "Deep video inpainting detection," 2021, *arXiv:2101.11080*.

[18] Y. Wu, W. AbdAlmageed, and P. Natarajan, "ManTra-Net: Manipulation tracing network for detection and localization of image forgeries with anomalous features," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9535–9544.

[19] C. Gao, A. Saraf, J.-B. Huang, and J. Kopf, "Flow-edge guided video completion," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 713–729.

[20] G. Boato, C. Pasquini, A. L. Stefani, S. Verde, and D. Miorandi, "TrueFace: A dataset for the detection of synthetic face images from social networks," in *Proc. IEEE Int. Joint Conf. Biometrics (IJCB)*, Oct. 2022, pp. 1–7.

[21] B. Zi, M. Chang, J. Chen, X. Ma, and Y.-G. Jiang, "WildDeepfake: A challenging real-world dataset for deepfake detection," in *Proc. 28th ACM Int. Conf. Multimedia*, Oct. 2020, pp. 2382–2390.

[22] C. Pasquini, I. Amerini, and G. Boato, "Media forensics on social media platforms: A survey," *EURASIP J. Inf. Secur.*, vol. 2021, no. 1, pp. 1–19, May 2021.

[23] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a," in *Proc. Adv. Neural Inf. Process. Syst.*, 1994, p. 737.

[24] W. Chen, X. Chen, J. Zhang, and K. Huang, "Beyond triplet loss: A deep quadruplet network for person re-identification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1320–1329.

[25] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol, and L. Bottou, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, no. 12, pp. 3371–3408, 2010.

[26] S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan, "YouTube-8M: A large-scale video classification benchmark," 2016, *arXiv:1609.08675*.

[27] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1725–1732.

[28] C. Galdi, F. Hartung, and J.-L. Dugelay, "SOCRatES: A database of realistic data for SOurce camera REcognition on smartphones," in *Proc. 8th Int. Conf. Pattern Recognit. Appl. Methods*, 2019, pp. 648–655.

[29] D. Shullani, M. Fontani, M. Iuliani, O. A. Shaya, and A. Piva, "VISION: A video and image dataset for source identification," *EURASIP J. Inf. Secur.*, vol. 2017, no. 1, pp. 1–16, Dec. 2017.

[30] Y. Zeng, J. Fu, and H. Chao, "Learning joint spatial–temporal transformations for video inpainting," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Aug. 2020, pp. 528–543.

[31] S. W. Oh, S. Lee, J.-Y. Lee, and S. J. Kim, "Onion-peel networks for deep video completion," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 4402–4411.

[32] Y. Wang, X. Tao, X. Qi, X. Shen, and J. Jia, "Image inpainting via generative multi-column convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 331–340.

[33] S. W. Oh, S. Lee, J.-Y. Lee, and S. J. Kim, "Supplementary material: Onion-peel networks for deep video completion," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2019, pp. 4403–4412.

[34] P. Yang, D. Baracchi, M. Iuliani, D. Shullani, R. Ni, Y. Zhao, and A. Piva, "Efficient video integrity analysis through container characterization," *IEEE J. Sel. Topics Signal Process.*, vol. 14, no. 5, pp. 947–954, Aug. 2020.

[35] M. Iuliani, D. Shullani, M. Fontani, S. Meucci, and A. Piva, "A video forensic framework for the unsupervised analysis of MP4-like file container," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 3, pp. 635–645, Mar. 2019.

[36] H. Li and J. Huang, "Localization of deep inpainting using high-pass fully convolutional network," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 8300–8309.

[37] F. J. Massey, "The Kolmogorov–Smirnov test for goodness of fit," *J. Amer. Stat. Assoc.*, vol. 46, no. 253, p. 68, Mar. 1951.

[38] B. Hadwiger and C. Riess, "The Forchheim image database for camera identification in the wild," in *Proc. Int. Conf. Pattern Recognit.*, Berlin, Germany, 2021, pp. 500–515. [Online]. Available: https://faui1-files.cs.fau.de/public/mmsec/datasets/fodb/

[39] D.-T. Dang-Nguyen, C. Pasquini, V. Conotter, and G. Boato, "RAISE: A raw images dataset for digital image forensics," in *Proc. 6th ACM Multimedia Syst. Conf.*, Mar. 2015, pp. 219–224.

[40] Q.-T. Phan, G. Boato, R. Caldelli, and I. Amerini, "Tracking multiple image sharing on social networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 8266–8270.

[41] S. Verde, C. Pasquini, F. Lago, A. Goller, F. De Natale, A. Piva, and G. Boato, "Multi-clue reconstruction of sharing chains for social media images," *IEEE Trans. Multimedia*, vol. 25, pp. 9491–9505, 2023.

DANIELE BARACCHI (Member, IEEE) received the bachelor's and master's degrees in computer engineering and the Ph.D. degree in information engineering from the University of Florence. Since 2018, he has been a member of the Image Analysis, Processing, and Protection Research Group, Department of Information Engineering, University of Florence. In this role, he is actively engaged in the development of machine learning-based techniques for multimedia forensics. He is currently a Postdoctoral Fellow with the University of Florence. Over the past four years, he has contributed to research initiatives supported by both the U.S. Defense Advanced Research Projects Agency (DARPA) and the Italian Ministry of University and Research (MUR).

GIULIA BOATO is currently an Associate Professor with the Department of Information Engineering and Computer Science, University of Trento, Italy. In 2012, she started a disruptive research direction on discrimination between virtual and real humans by exploiting signal processing, analysis of physiological signals, and more recently cutting edge deep learning techniques. This also included the detection of various types of manipulation of digital media, recently focusing on the detection of deepfakes and on forensic analysis in an open world scenario where data is shared on social media. She is the author of more than 140 papers in international conferences and journals, with an H-index of 30. Her research interests include image and signal processing, with particular attention to multimedia data protection and digital forensics. She is an elected member of the IEEE Multimedia Signal Processing Technical Committee (MMSP TC) and the IEEE Information Forensics and Security Technical Committee (IFS TC).

FRANCESCO DE NATALE (Senior Member, IEEE) received the M.Sc. and Ph.D. degrees, in 1990 and 1994, respectively. He is currently a Professor of telecommunications with the DISI, University of Trento, Italy, leading the Multimedia Technologies Laboratory (MMLab). He is also the Director of CNIT, Italian National Consortium of Telecommunications. He published more than 250 scientific works on international conferences. He also contributed to numerous research and development projects, funded by different national and international agencies. His research interests include the various aspects of multimedia signal processing, analysis and communications, including semantic multimedia retrieval, image forensics, and smart multisensory environments. He is a member of ACM and CVPL. He was the TP Co-Chair of IEEE ICIP, in 2005. He was the Co-Founder and the General Chair of ACM ICMR, in 2011. He was an Associate Editor of IEEE TRANSACTIONS ON MULTIMEDIA and IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY. He is a Senior Associate Editor of IEEE TRANSACTIONS ON IMAGE PROCESSING.

MASSIMO IULIANI received the master's degree in applied mathematics from the University of Florence. He is currently a Postdoctoral Fellow with the University of Florence. He is also with the Image Analysis Processing and Protection Group, Department of Information Engineering, University of Florence. He is also a Technical Supervisor with the FORLAB, the Multimedia Forensics Laboratory, University of Florence. In the last seven years, he worked on research projects funded by the European Commission (EC) and the U.S. Defense Advanced Research Projects Agency (DARPA). All projects were related to authentication and reverse engineering of multimedia contents. His main activities involve the training of law enforcement and legal operators and the consultancy multimedia contents analysis (digital images, audio and videos) for forensic purposes.

ANDREA MONTIBELLER received the B.Sc. and M.Sc. degrees in information and communications engineering from the University of Trento, Italy, in 2018 and 2020, respectively, where he is currently pursuing the Ph.D. degree in information engineering and computer science. In 2018, he was a Visiting Researcher with the Department of Signal Theory and Communications, University of Vigo, Spain. In 2023, he was a Research Scholar with the Department of Electrical and Computer Engineering, Drexel University, Philadelphia, PA, USA. His research interests include AI generated image and video forgery detection and localization and camera source attribution.

CECILIA PASQUINI received the Ph.D. degree in information and communication technology from the University of Trento, Italy, in 2016. She is currently a Researcher with Fondazione Bruno Kessler (FBK), Italy. Prior to that, she has been a Researcher with the MMLab, University of Trento, and the Privacy and Security Laboratory, Universität Innsbruck, Austria. Her research interests include information security and image/video processing, with a focus on media security and forensics, synthetic media detection, and adversarial machine learning. She has participated in several projects on these topics and coauthored numerous scientific publications. She is an elected member of the EURASIP BForSec Technical Area Committee. She is the Program Chair of ACM IH&MMSec 2023. She serves as a reviewer for many journals and conferences in the field.

ALESSANDRO PIVA (Fellow, IEEE) is currently an Associate Professor with the Department of Information Engineering, University of Florence. He is also the Head of the FORLAB, the Multimedia Forensics Laboratory, University of Florence. His research interests include information forensics and security, and image and video processing. In the first topic, he was interested in data hiding, signal processing in the encrypted domain, and image and video forensic techniques. In the second area, he was interested in the design of image and video processing and analysis techniques for cultural heritage, medical, and industrial applications. In the above research topics, he is the coauthor of more than 50 articles published in international journals and 120 papers published in international conference proceedings, with an H-index of 40 according to Scopus.

DASARA SHULLANI (Member, IEEE) received the master's degree in computer engineering from Politecnico di Torino and the Ph.D. degree in information engineering from the University of Florence. Since 2015, she has been a member of the Image Analysis, Processing, and Protection Research Group, Department of Information Engineering, University of Florence, where she is developing multimedia forensics tools applied to video contents. She is currently a Postdoctoral Fellow with the University of Florence. During this period, she has worked on research projects funded by the Consortium GARR, the U.S. Defense Advanced Research Project Agency (DARPA), and the Italian Ministry of University and Research (MUR).

Open Access funding provided by 'Università degli Studi di Firenze' within the CRUI CARE Agreement