Multi-clue reconstruction of sharing chains for social media images

Sebastiano Verde, Cecilia Pasquini, Federica Lago, Alessandro Goller,
Francesco De Natale, Alessandro Piva, and Giulia Boato

arXiv:2108.02515v1 [cs.MM] 5 Aug 2021

Abstract—The amount of multimedia content shared everyday, combined with the level of realism reached by recent fake-generating technologies, threatens to impair the trustworthiness of online information sources. The process of uploading and sharing data tends to hinder standard media forensic analyses, since multiple re-sharing steps progressively hide the traces of past manipulations. At the same time though, new traces are introduced by the platforms themselves, enabling the reconstruction of the sharing history of digital objects, with possible applications in information flow monitoring and source identification. In this work, we propose a supervised framework for the reconstruction of image sharing chains on social media platforms. The system is structured as a cascade of backtracking blocks, each of them tracing back one step of the sharing chain at a time. Blocks are designed as ensembles of classifiers trained to analyse the input image independently from one another by leveraging different feature representations that describe both content and container of the media object. Individual decisions are then properly combined by a late fusion strategy. Results highlight the advantages of employing multiple clues, which allow accurately tracing back up to three steps along the sharing chain.

I. INTRODUCTION

MASSIVE amounts of multimedia data are uploaded every day to social media platforms by nearly 4 billion active users: according to recent estimates, 3.2 billion images are shared every day [1] and 500 hours of video are uploaded to YouTube every minute [2]. At the same time, easy-to-use editing tools that allow modifying such multimedia data have become widely available. While this enables for an unprecedented ease in sharing information, it also entails serious implications for the trustworthiness and reliability of digital media.

Such concerns reached a critical level with the recent development of tools based on artificial intelligence (AI) that allow even inexperienced users generating almost automatically highly realistic fakes, especially when dealing with images and videos depicting faces [3], [4]. Indeed, advanced tools like AI and photo/video editing used to be restricted to skilled users and researchers but are nowadays available to a much wider public, likely going beyond the primary purpose of entertainment. As for any other technology, a reasonable risk exists for malicious misuse, e.g., conveying misinformation to bias people and influence social groups [5]. Moreover,

S. Verde, C. Pasquini, F. Lago, A. Goller, F. De Natale and G. Boato are with the *Dipartimento di Ingegneria e Scienza dell'Informazione* (DISI), University of Trento, Italy.

A. Piva is with the *Dipartimento di Ingegneria dell'Informazione* (DINFO), University of Florence.

multimedia data are responsible for the viral diffusion of information through social media and web channels, and play a key role in the digital life of individuals and societies. Therefore, developing tools to preserve the trustworthiness of shared images and videos is a necessary step that our society can no longer ignore.

Several works in media forensics have investigated the detection of manipulations and the identification of the digital source, providing promising results in laboratory condition and well-defined scenarios [6]. More recently, the research community has also pursued the ambition to scale forensic analyses to real-world web-based systems, which involve routinely applied operations such as the act of sharing through social media platforms [7]. This extension requires the ability to face significant technological challenges related to the (possibly multiple) uploading/sharing processes, and hence the need for methods that can reliably work under more general conditions. Retrieving information about the life of a digital object in terms of provenance, manipulations and sharing operations, would indeed represent a valuable asset: on one hand, it could support law enforcement agencies and intelligence services in tracing perpetrators of deceptive visual contents; on the other hand, it could help in preserving the trustworthiness of digital media and countering the effects of misinformation.

The process of uploading to web platforms represents nowadays a key phase in the life of a digital object, and the dynamics of shared visual content can be analyzed for different purposes [8], [9]. While this sharing process typically hinders the ability to perform conventional media forensics tasks, it also introduces new traces itself, allowing to infer additional information. As a matter of fact, data can be uploaded in different ways, multiple times, on diverse platforms and from different systems. In this context, the possibility of reconstructing the sharing history of a given object, known as *platform provenance analysis* [7], could help monitoring the information flow by tracing back previous uploads, and thus supporting source identification by narrowing down the search.

Distinct traces are left on a digital image when uploaded to a web platform or a social network, depending on the operations that are involved in the process. As firstly observed in the case of Facebook [10], compression and resizing are typically applied to reduce the size of uploaded images, and this is performed differently on different platforms and depending on the resolution of the original image. As known in media forensics, such operations can be detected and characterized by analyzing the image content or signal (i.e., the values in the pixel domain or in various transformed domains). A signal-

Fig. 1. Visual representation of the media recycling problem where we aim at reconstructing the sharing history of a given image. In this work we consider up to three sharing steps on three different platforms, namely Facebook (FB), Flickr (FL) and Twitter (TW). Three examples of reconstruction of the sharing history are presented: the red one has been shared first on TW and then on FB; the green one on TW, FB and FL; the yellow only on TW (note that the reconstruction proceeds backwards, starting from the most recent sharing step).

based approach for platform provenance analysis can leverage the Discrete Cosine Transform (DCT) coefficients [11], [12], the PRNU noise [13] or combinations of the two [14]. Moreover, provenance can also be inferred as a by-product of detectors developed to analyse image manipulations in the pixel domain, as shown in [15].

Meaningful information can be extracted as well from the image container, such as metadata. While signal-based forensic analysis is usually preferable (as data structures can be erased or falsified more easily than signals), such clues play a relevant role in platform provenance analysis, being typically related to the platform itself rather than to the acquisition device [16]. In [17] the authors consider several popular platforms (namely Facebook, Google+, Flickr, Tumblr, Imgur, Twitter, WhatsApp, Tinypic, Instagram, Telegram), observing two main facts: (i) the uploaded files are renamed with distinctive patterns which can even allow to reconstruct the file's web URL; and (ii) both resizing and JPEG compression adopt platform-specific rules. Therefore, the authors propose a feature representation that includes image resolution and quantization table coefficients, which can be extracted from the file without decoding.

An even more challenging scenario consists in the reconstruction of sharing chains, meaning that the target of the analysis goes beyond the identification of the latest sharing platform and aims at reconstructing the history of a digital object that has been re-shared multiple times, possibly on different platforms. We denote this scenario as *media recycling*. Since forensic traces tend to decay as we keep applying new operations on an image, the problem of media recycling requires the combination of multiple detection strategies to be solved. Hybrid approaches where clues are extracted from both signal and metadata have been proposed in [18], [19], [20], showing promising results in the identification of sharing steps beyond the most recent one.

In this paper we address the image recycling problem by proposing a novel multi-clue detection system able to reconstruct the sharing history of images on various platforms. The proposed framework is designed as a cascaded architecture, which allows tracing back one sharing platform at a time, leveraging the knowledge of the previously detected steps to reconstruct the whole sequence step by step (Figure 1). The system employs an effective fusion mechanism to combine multiple classifiers, thus allowing to successfully exploit both content and container of the inspected image, and being open to possible integration with other sets of features. As an additional contribution, we present a novel set of container-related features extracted from the JPEG header, which allows significantly boosting performances when combined to other detectors.

The proposed framework is evaluated on a published dataset of images shared on different platforms. Experiments, conducted including both single detectors and fused ones, show that the combination of heterogeneous traces is beneficial in media recycling detection. Thanks to the novel set of container-based features, the identification of the last sharing step reaches 100% accuracy on the test data. Moving further back in the history, the system allows reconstructing the sharing chain with high accuracy up to the third step.

A repository with the implementation of the presented system is publicly available[1].

The paper is structured as follows: Section II formally defines the image recycling problem and describes the architecture of the proposed reconstruction system; Section III presents a set of forensic traces that provide meaningful information in the context of the image recycling scenario, along with the adopted strategy for fusing such multiple

[1] https://github.com/Flake22/sharing-chains-reconstruction

descriptors; Section IV discusses the experimental setting and the obtained results, and reports a feature separability analysis for the presented descriptors; finally, Section V draws the conclusions.

II. MULTI-STEP RECONSTRUCTION OF SHARING CHAINS

The problem of image recycling involves media data that have been shared one or more times through (possibly different) social media and web platforms.

The key assumption is that a given image $\mathbf{x} \in \mathcal{I}$ underwent a number $\ell \geq 1$ of sharing steps, forming a *sharing chain*.

Definition 1: Given a set $\mathcal{S}$ of sharing platforms, a *sharing chain* of length ℓ is a sequence of sharing platforms, which we can be represented as a vector $\mathbf{C}$ in $\mathcal{S}^\ell \doteq \underbrace{\mathcal{S} \times \ldots \times \mathcal{S}}_{\ell}$.

For the sake of convenience, we will index the components of $\mathbf{C}$ in reverse order, so that the temporal succession of sharing platforms goes as follows:

$$\mathbf{C}[-(\ell-1)] \to \ldots \to \mathbf{C}[-1] \to \mathbf{C}[0], \quad (1)$$

and thus $\mathbf{C}[0]$ corresponds to the last sharing step in $\mathbf{C}$.

Definition 2: For a generic chain $\mathbf{C} \in \mathcal{S}^\ell$, the set $\mathcal{B}(\mathbf{C})$ contains all the chains of length $\ell+1$ whose last ℓ components are identical to the ones of $\mathbf{C}$; in other words, chains in $\mathcal{B}(\mathbf{C})$ differ from $\mathbf{C}$ by one additional previous sharing step, $\mathbf{C}[-\ell]$.

Let us also define Ω_ℓ as the set of all possible sharing chains of length up to ℓ obtained through the combination of platforms in $\mathcal{S}$,

$$\Omega_\ell \doteq \bigcup_{1 \leq i \leq \ell} \mathcal{S}^i \quad (2)$$

In our formulation, the goal of a recycling analysis is to devise a system $\mathcal{F}_L$ that assigns a certain image $\mathbf{x}$ to the sharing chain it went through, up to a predefined maximum chain length L.

A. Cascaded architecture

Given a predefined number L of sharing steps, our purpose is to devise a system $\mathcal{F}_L : \mathcal{I} \to \Omega_L$ that takes in input an image $\mathbf{x}$ and associates it to the correct chain of maximum length L.

In our approach, we propose to structure $\mathcal{F}_L$ as a multi-step cascade of *backtracking blocks* $F_{-\ell}$, each of them tracing back one step of the sharing chain at a time.

In particular, we define:

- $F_0 : \mathcal{I} \to \Omega_1 \equiv \mathcal{S}$
  This first backtracking block assigns $\mathbf{x}$ to the platform in $\mathcal{S}$ that corresponds to the last sharing step.

- $F_{-\ell} : \mathcal{I} \times \Omega_\ell \to \Omega_{\ell+1}, \quad \ell = 1, \ldots, L-1$
  For $\ell > 1$, based on the knowledge of the previous blocks' decisions, each backtracking block inspects a possible preceding sharing operation. This process is recursively performed until ℓ reaches $L-1$, and the last block provides the final output.

Formally, we can express the whole system as

$$\mathcal{F}_L(\mathbf{x}) = F_{-(L-1)}(\mathbf{x}, \ldots, F_{-2}(\mathbf{x}, F_{-1}(\mathbf{x}, F_0(\mathbf{x})) \ldots) \quad (3)$$

By defining the intermediate chains $\mathbf{C}_{-\ell}$ as the output of $F_{-\ell}$, i.e., $\mathbf{C}_{-\ell} \doteq F_{-\ell}(\mathbf{x}, F_{-(\ell-1)}(\mathbf{x}))$, we can represent the multi-step process as in Figure 2.

Fig. 2. Schematic representation of the multi-step cascade. Backtracking blocks are represented in squares; black dashed lines indicate input arguments to the blocks, solid red lines the output of the blocks.

Each intermediate backtracking block assigns $\mathbf{x}$ to a potentially longer chain, in the case a previous sharing step is detected. This is done through an ensemble of classifiers, each one specialized in dealing with a specific output of the previous block. If no additional steps are detected, the intermediate chain does not increase in length and is regarded as the final output. By dropping the subscript for the sake of simplicity and indicating as $\mathbf{C} \in \Omega_\ell$ an arbitrary intermediate chain determined at the previous backtracking block, we can formulate a generic intermediate block $F_{-\ell}$ as follows:

$$F_{-\ell}(\mathbf{x}, \mathbf{C}) = \begin{cases} \mathbf{C} & \text{if } \mathbf{C} \in \Omega_{\ell-1} \\ f_{-\ell}^{\mathbf{C}}(\mathbf{x}) & \text{if } \mathbf{C} \in \Omega_\ell \setminus \Omega_{\ell-1} \end{cases} \quad (4)$$

The first case corresponds to the situation where no backward step is detected at the previous backtracking block (i.e., the length of $\mathbf{C}$ is lower than ℓ), therefore no additional steps should be added to the current chain. In the second case, the functions $f_{-\ell}^{\mathbf{C}}(\cdot)$ are specialized detectors trained to disambiguate among the set containing the chain $\mathbf{C}$ and all the ones that include a previous sharing step. Thus, by design, one detector $f_{-\ell}^{\mathbf{C}} : \mathcal{I} \to \{\mathbf{C}\} \cup \mathcal{B}(\mathbf{C})$ is needed for each $\mathbf{C} \in \Omega_\ell \setminus \Omega_{\ell-1}$. By design, we indicate $f_0(\mathbf{x}) \equiv F_0(\mathbf{x})$.

The way specialized detectors assign $\mathbf{x}$ to either $\mathbf{C}$ or a longer chain involves the combination of different recycling traces. The description of the employed traces and the fusion technique adopted to combine the associated detectors is the subject of the next section.

For the sake of clarity, we report in Figure 3 a visual example of how the system $\mathcal{F}_L$ works for the case $L = 3$ and $\mathcal{S} = \{\text{FB}, \text{TW}, \text{FL}\}$, which refer to Facebook, Twitter and Flickr, respectively.

III. EXTRACTION AND FUSION OF RECYCLING TRACES

Traces of media recycling can be found on image data under investigation exploiting different domains. In our case, we define multiple feature representations extracted from both image signal and image container. As far as signal-based features are

Fig. 3. Visual example of a run of the system $\mathcal{F}_L$, with $L = 3$ and $\mathcal{S} = \{\text{FB}, \text{TW}, \text{FL}\}$. Backtracking blocks are annotated in gray on the left; black dashed lines represent arguments that are given in input to specialized detectors, while solid red lines indicate the outputs of specialized detectors. In this case, the chain (TW, FB) is reconstructed: after $F_0(\mathbf{x})$ identifies Facebook as last sharing platform $\mathbf{C}_0$, the block $F_{-1}(\mathbf{x}, \mathbf{C}_0)$ calls the specialized detector $f_{-1}^{(\text{FB})}(\mathbf{x})$, which detects a previous sharing on Twitter and returns the chain $\mathbf{C}_{-1} = (\text{TW}, \text{FB})$; then, the block $F_{-2}(\mathbf{x}, \mathbf{C}_{-1})$ calls the specialized detector $f_{-2}^{(\text{TW}, \text{FB})}(\mathbf{x})$, which returns again the chain $\mathbf{C}_{-2} = (\text{TW}, \text{FB})$, thus fixing the chain length to 2. The final output of the system is therefore $\mathcal{F}_3(\mathbf{x}) = \mathbf{C}_{-2} = (\text{TW}, \text{FB})$.

Fig. 4. Example of JPEG header representation as extracted by ExifTool. In this case, the image has been shared once through Twitter.

concerned, we focus on the histograms of DCT coefficients (denoted as **DCT** and described in Section III-A), which have been successfully adopted in the literature to address several media forensics problems [18], [21], [22]. Regarding container-based information, two different feature vectors are defined: **META**, which encodes metadata mostly related to the last JPEG compression settings, and **HEADER**, which contains information on the overall structure of the JPEG header; these feature descriptors are described in Section III-B. In particular, the latter represents a recycling trace that we propose for the first time in this work and is capable of boosting the overall performance when combined to other features.

A. Content-based features

The set of content-based features, denoted as **DCT**, encodes the forensic traces left on the image signal by the lossy compression of JPEG standard, which is currently the most common format for images uploaded on social networks. The specific processing, however, is peculiar to the needs of each platform. In particular, the target image quality is controlled by the integer quantization in the DCT domain, which is reflected in the statistics of the quantized coefficients.

Following the scheme in [11], the 8×8 block-based DCT is first computed on the whole image; then, normalized histograms of dequantized DCT coefficients are extracted from the first 9 AC frequencies, in zigzag order. We retained for each histogram the 41 bins corresponding to the range of integers between -20 and 20. Finally, the concatenation of the histograms provides a 369-dimensional feature vector.

B. Container-based features

The first set of container-based features, denoted as **META**, exploits the information about the JPEG compression settings contained in the metadata of the image under investigation. These features, as proposed in [19], consist of a 152-dimensional vector encoding the following information:

- *Quantization tables* (128), for luminance and chrominance channels, reflecting the JPEG quality factor
- *Huffman encoding tables* (2), number of tables used for AC and DC component
- *Component information* (18), describing component id, horizontal/vertical sampling factors, quantization table index and AC/DC coding table indices, for each YCbCr component
- *Optimized coding and progressive mode* (2), binary features indicating the use of the two modes
- *Image size* (2), the image dimensions

The second set of features, denoted as **HEADER**, is a novel contribution of this work and encodes structural properties of the JPEG header.

Previous approaches in [23], [16] observed that the EXIF information of JPEG files can contain useful information for forensics purposes, such as authentication and source identification. More recently, authors in [24] proposed an efficient container-based method to verify the integrity of videos, showing also the possibility to identify the social media platform on which the video was uploaded.

A similar idea is exploited in this work, adapted to the image recycling scenario. In fact, while sharing platforms usually strip out optional metadata fields (like acquisition time and device, GPS coordinates, etc.), we observed that different platforms retain different EXIF fields in the downloaded JPEG files, which can be extracted through several possible tools and be employed as feature descriptors.

Our implementation is based on ExifTool [25], a free and open-source software library for reading, writing, and editing image containers. In particular, ExifTool encodes the header of a JPEG file into an HTML page, as exemplified in Figure 4.

Every extracted file contains header markers indicating the beginning of a specific kind of segment. Those can be found

Fig. 5. Comparison of **HEADER** features extracted from the same image shared on different platforms.

multiple times throughout the header, and their frequency represents a discriminative property for detecting the upload on a sharing platform.

For this purpose, a set of 8 markers was selected, as detailed below:

- *DHT*: encapsulates the information regarding the Huffman Table
- *unused*: unused data blocks for maintaining a fixed size
- *APP13*: provides a number of methods for managing Photoshop/IPTC data without dealing with the details of the low level representation
- *APP2*: used to store the ICC profile
- *SOF0*: Start Of Frame (Baseline DCT), indicates a baseline DCT-based JPEG, and specifies the width, height, number of components, and component subsampling
- *SOF2*: Start Of Frame (Progressive DCT), indicates a progressive DCT-based JPEG, and specifies the width, height, number of components, and component subsampling
- *cmp3*: comment section
- *JPEG DRI*: Define Restart Interval.

For each image file, the frequency of the listed segments is computed throughout the JPEG header, providing a 8-element feature vector. An example is reported in Figure 5, where the feature vector is extracted for the same image when subject to a sharing operation on different platforms. The resulting **HEADER** descriptor is therefore low-dimensional and does not require to decode the image, since information are read directly from the file header. In addition to being computationally efficient, **HEADER** provides an extremely accurate identification of the last sharing step, and remains informative enough to boost the performance of other descriptors in further steps of the chain, as demonstrated in Section IV.

C. Fusion of classifiers

The described feature representations can be employed to train specific classifiers that discriminate among pre-defined sets of sharing chains. However, in order to implement the functions $f_{-\ell}^{\mathbf{C}}(\mathbf{x})$ described in (4), we need to combine the output of such classifiers into a single overall decision. For easier reading, from now on we will simplify the notation by removing any reference to the block index ℓ and the previous

output $\mathbf{C}$, so that we can discuss the implementation of a generic function $f(\mathbf{x})$.

The above scenario is typically referred to in the literature as *combination of multiple experts* (CME) [26], [27], where each classifier is regarded as an "expert" who analyses one specific aspect of the object under investigation, and the final response is obtained by properly merging all the individual decisions.

Formally, the CME problem involves K experts (or classifiers) denoted by e_k, $k = 1, \ldots, K$, which share the same set of mutually exclusive output classes. The function $f(\mathbf{x})$ combines the individual decisions (the outputs of the K classifiers) by means of a fusion function g, and provides an overall classification,

$$f(\mathbf{x}) = g\left(e_1(\mathbf{x}), e_2(\mathbf{x}), \ldots, e_K(\mathbf{x})\right) \qquad (5)$$

Note that the individual decisions and the combined one all belong to the same set of chains (which depends on the block index and the result of the previous block). Regardless of the specific set, however, the number of chains is always $|\mathcal{S}| + 1$, since we can either add a new step from $\mathcal{S}$ to the partial chain or not.

The CME problem, i.e., the implementation of the fusion function g, is addressed in our work by means of a method known as *behavior-knowledge space* (BKS) [28], [29], which allows building prior knowledge about the behavior of the individual classifiers without requiring the statistical independence of the same.

A BKS is a K-dimensional space where each dimension is related to the decisions of one classifier. Each classifier has the same number of decisions, chosen from a given set, and the intersection of the decisions of individual experts identifies a *unit* of the BKS.

Given an input $\mathbf{x}$, for which we have $e_k(\mathbf{x})$, $k = 1, \ldots, K$, i.e., the individual decisions, the corresponding unit in the BKS has coordinates $(e_1(\mathbf{x}), e_2(\mathbf{x}), \ldots, e_K(\mathbf{x}))$ and is called the *focal unit* for $\mathbf{x}$. Given that, in our case, the number of output classes for each classifier is $|\mathcal{S}| + 1$, it follows that the BKS contains exactly $Q = (|\mathcal{S}| + 1)^K$ units, denoted by u_q, $q = 1, 2, \ldots, Q$.

In practice, the BKS is implemented as a lookup table that associates each unit (i.e., each combination of individual decisions) to a probability distribution over the set of output classes. Such distributions are estimated by accumulating the number of incoming samples for each class on a dedicated training set (different from the one used to train the individual classifiers [30]).

Let us define $y \in \{1, 2, \ldots, |\mathcal{S}| + 1\}$, a set of numerical labels associated to the output classes. A representation of a BKS lookup table is given in Table I: each unit u_q, $q = 1, \ldots, Q$, corresponds to a unique combination of classifier decisions, and $n_y^{(q)}$ is the number of training samples belonging to class y that have fallen into the q-th unit.

Let us assume that the input $\mathbf{x}$ is mapped into a set of decisions $e_k(\mathbf{x})$, $k = 1, \ldots, K$, that corresponds to the focal unit u_q of the BKS. From the lookup table, we can derive the

Fig. 6. Framework architecture for multi-clue reconstruction of sharing chains. The input image $\mathbf{x}$ is fed into a cascade of backtracking blocks, $F_{-\ell}$, each one dedicated to identifying chains of length up to $\ell + 1$. Every step consists of an ensemble of parallel classifiers, which analyse $\mathbf{x}$ by also taking into account the information from previous steps, and a fusion module that provides the overall decision. Stopping conditions intervene when $F_{-\ell}$ receives a chain of length $\ell - 1$, i.e., the end of the chain has been reached.

TABLE I
BKS LOOKUP TABLE.

| | BKS units | | | | |
	u_1	u_2	$\ldots$	u_q	$\ldots$	u_Q
Class y	$n_1^{(1)}$	$n_1^{(2)}$	$\ldots$	$n_1^{(q)}$	$\ldots$	$n_1^{(Q)}$
	$n_2^{(1)}$	$n_2^{(2)}$	$\ldots$	$n_2^{(q)}$	$\ldots$	$n_2^{(Q)}$
	$\vdots$	$\vdots$		$\vdots$		$\vdots$
	$n_{|\mathcal{S}|+1}^{(1)}$	$n_{|\mathcal{S}|+1}^{(2)}$	$\ldots$	$n_{|\mathcal{S}|+1}^{(q)}$	$\ldots$	$n_{|\mathcal{S}|+1}^{(Q)}$

posterior probability of associating input $\mathbf{x}$ to a label y given the focal unit u_q as follows

$$P\left(f(\mathbf{x}) = y \mid u_q\right) = \frac{n_y^{(q)}}{\sum_{j=1}^{|\mathcal{S}|+1} n_j^{(q)}} \quad (6)$$

The optimal combined decision is therefore

$$\hat{y} = \underset{y}{\arg\max}\, P\left(f(\mathbf{x}) = y \mid u_q\right) \quad (7)$$

In BKS fusion, there is also the possibility of an input $\mathbf{x}$ being *rejected*, meaning that while the individual decisions are valid their combination is impossible. In practice, a rejection occurs when the focal unit is empty, i.e., no training samples have been mapped into u_q, or when the maximum value of the estimated distribution $n_y^{(q)}$ is non-unique.

In the cascade architecture, a rejection from the fusion module stops the reconstruction process seamlessly, since its effect is equivalent to the case when the end-point of a sharing chain is reached (see Equation 4).

Figure 6 depicts a comprehensive representation of the cascaded architecture, exemplified by two backtracking blocks, including the fusion modules and the stopping conditions; the classifiers related to the three adopted descriptors (DCT, META, HEADER) are denoted by e_D, e_M, e_H.

IV. EXPERIMENTS AND RESULTS

A. Experimental setting

The presented framework is highly flexible, being adaptable to a different number K of feature classifiers, set $\mathcal{S}$ of sharing platforms, and length L of the longest detectable sharing chain.

In our experiments, we implemented the system with the following configuration:

- $K = 3$ feature descriptors (**DCT**, **META**, **HEADER**);
- $\mathcal{S} = \{\text{FB}, \text{FL}, \text{TW}\}$, where the labels denote Facebook, Flickr and Twitter, respectively;
- $L = 3$, therefore the system $\mathcal{F}_L$ is composed of three backtracking blocks, F_0, F_{-1}, F_{-2}.

It follows that the set of all possible sharing chains identified by the system is $\Omega_3 = \mathcal{S} \cup \mathcal{S}^2 \cup \mathcal{S}^3$, where:

- $\mathcal{S}^2 = \{(\text{FB}, \text{FB}), (\text{FL}, \text{FB}), (\text{TW}, \text{FB}), \ldots\}$, $|\mathcal{S}^2| = 9$;
- $\mathcal{S}^3 = \{(\text{FB}, \text{FB}, \text{FB}), (\text{FL}, \text{FB}, \text{FB}), \ldots\}$, $|\mathcal{S}^3| = 27$;

and thus $|\Omega_3| = 39$.

1) Dataset description: the framework was trained and tested on the **R-SMUD** dataset [19], containing images shared on the three social platforms in $\mathcal{S}$. Images are generated starting from the RAISE dataset [31] by cropping 50 raw format images on the top-left corner, with fixed aspect ratio of 9:16 and different resolutions: 377x600, 1012x1800 and 1687x3000. Each of the obtained crops was then compressed with six quality factors (50, 60, 70, 80, 90, 100) and then shared up to a maximum of 3 times through different combinations of the three platforms in $\mathcal{S}$, providing a total of 35100 images and $|\Omega_3| = 39$ classes. The dataset was then divided into training, validation and test subsets, with a 60/20/20 split.

2) Classifiers and training: as described in Section III-C, each backtracking block contains an ensemble of trained classifiers, one for each combination of feature descriptor and output of the previous block.

Given the number K of feature descriptors, and given that one detector $f_{-\ell}^{\mathbf{C}}$ is needed for each partially reconstructed

chain $\mathbf{C} \in \Omega_\ell \setminus \Omega_{\ell-1}$ (see Section II-A), we can derive the number of classifiers needed at each backtracking block:

- $F_0 \longrightarrow K = 3$
- $F_{-1} \longrightarrow K \cdot |\Omega_1| = 9$
- $F_{-2} \longrightarrow K \cdot |\Omega_2 \setminus \Omega_1| = 27$

We also recall from Section III-C that the number of output classes of each detector, regardless of the step index ℓ and the partial chain $\mathbf{C}$, is equal to $|\mathcal{S}| + 1 = 4$. For instance, if $F_0(\mathbf{x}) = \mathbf{C}_0 = \text{FB}$, then the output of $F_{-1}(\mathbf{x}, \mathbf{C}_0)$ can either be $\mathbf{C}_{-1} = \text{FB}$ or $\mathbf{C}_{-1} = (*, \text{FB}) \in \mathcal{B}(\mathbf{C}_0)$, where $*$ indicates whatever element in $\mathcal{S}$.

Each detector was implemented as a Random Forest Classifier (RFC), with fixed hyper-parameters. In particular, we used a number of estimators equal to 10, balancing accuracy and model complexity.

Classifiers were trained on the training subset of R-SMUD, containing 21060 images equally distributed among the 39 classes. Detectors belonging to F_0 and F_{-1}, which are dedicated to the classification of shorter chains than F_{-2}, were trained after re-mapping the original 39 classes as follows:

- F_0 is trained on chains in Ω_1,

$$(\mathbf{C}[-2], \ \mathbf{C}[-1], \ \mathbf{C}[0]) \in \Omega_3 \longrightarrow \mathbf{C}[0] \in \Omega_1;$$

- F_{-1} is trained on chains in Ω_2,

$$(\mathbf{C}[-2], \ \mathbf{C}[-1], \ \mathbf{C}[0]) \in \Omega_3 \longrightarrow (\mathbf{C}[-1], \ \mathbf{C}[0]) \in \Omega_2.$$

Such training strategy allows fully exploiting the amount of samples in the dataset and at the same time recreating a more realistic scenario: as an example, the detection of $\mathbf{C}[0]$ at step F_0 is carried out among chains of different lengths, having $\mathbf{C}[0]$ as their last step.

Since steps $F_{-\ell}$, $\ell > 0$, discriminate chains that are known up to $\mathbf{C}[-(\ell-1)]$, the related classifiers need to be trained on smaller subsets of samples, obtained by fixing $\mathbf{C}[0], \mathbf{C}[-1], \ldots, \mathbf{C}[-(\ell-1)]$. In general, the fraction of training samples available to train step $F_{-\ell}$ is $1/|\Omega_\ell \setminus \Omega_{\ell-1}|$. For instance, at step F_{-1} we have to split the training according to the possible outputs of F_0, and thus we get $1/|\Omega_1| = 1/3$, corresponding to 7020 images; similarly, at step F_{-2} we get $1/|\Omega_2 \setminus \Omega_1| = 1/9$, corresponding to 2340 images.

As for the fusion part, the number of BKS modules required at each backtracking block equals the number of classes output by the related previous block, similarly to the classifiers. Each fusion module was trained on the validation subset of R-SMUD (7020 images) to prevent biasing the interaction between classifiers and fusions (Section III-C). The same dataset partitioning described above for steps $F_{-\ell}$, $\ell > 0$, was carried out for the training of fusion modules.

Finally, the end-to-end system was evaluated on the test set, containing 7020 images. The experimental results are reported and commented in the next paragraphs.

B. Reconstruction results

Each step of the cascade architecture was evaluated separately, in order to observe the reconstruction behavior throughout the pipeline. In detail, we ran the system on the test set of R-SMUD and measured the detection accuracy at the output

TABLE II
PER-STEP PERFORMANCE OF THE CASCADE SYSTEM.

	F_0 3 classes		F_{-1} 12 classes		F_{-2} 39 classes	
Single classifiers	ACC		ACC		ACC	
DCT	0.8634		0.4620		0.1849	
META	0.9296		0.5350		0.2959	
HEADER	**1.0000**		0.5994		0.2289	
Random guess	0.3333		0.0833		0.0256	
Fused classifiers	ACC	REJ	ACC	REJ	ACC	REJ
DCT+META	0.9296	0.0000	0.6096	0.0198	0.3475	0.1849
META+HEADER	**1.0000**	0.0000	0.7934	0.0188	**0.5576**	0.2325
DCT+META+HEADER	**1.0000**	0.0000	**0.7981**	0.0208	0.5465	0.1949

Fig. 7. Detection performance of backtracking block F_0, which identifies the last platform in the sharing chain, $\mathbf{C}[0]$; tests carried out with individual feature sets (a–c), pair fusions (d–e) and triplet fusion (f).

of each backtracking block, $F_{-\ell}$, $\ell = 0, 1, 2$. We recall that the number of output classes at each block is equal to $|\Omega_{\ell+1}|$, hence: 3 classes at F_0; 12 classes at F_{-1}; 39 classes at F_{-2}.

Additionally, we ran the experiments by including different subsets of feature descriptors in the system, to better assess the individual contributions: first, we tested **DCT**, **META** and **HEADER** features by themselves; then, the fusion of pairs of features; and finally, the fusion of all the three.

Table II reports an overview of accuracy values for each step of the cascade and for all feature configurations; for fused classifiers, we also report the rejection rates, i.e., the percentage of test samples rejected by the BKS fusion. At a macroscopic level, we can observe the following: (i) fused classifiers perform better than (or on par with) single ones, suggesting that the usage of heterogeneous feature descriptors is beneficial for platform provenance analysis; (ii) **HEADER** features allow a *deterministic* identification of the last sharing platform, which is preserved even when fused with other classifiers; (iii) the overall accuracy values tend to decrease as we proceed through the cascade, while rejection rates increase.

To deeper investigate these preliminary observations, the following paragraphs illustrate in details, with the aid of con-

(a) **DCT**, F_{-1}.

(b) **META**, F_{-1}.

(c) **HEADER**, F_{-1}.

(d) **DCT+META**, F_{-1}.

(e) **META+HEADER**, F_{-1}.

(f) **DCT+META+HEADER**, F_{-1}.

Fig. 8. Detection performance of backtracking block F_{-1}, which identifies the last two platforms in the sharing chain, $(\mathbf{C}[-1], \mathbf{C}[0])$; tests carried out with individual feature sets (a–c), pair fusions (d–e) and triplet fusion (f). Red squares highlight the subsets of classes having $\mathbf{C}[0]$ in common; note how most confusions are confined within the diagonal squares (all of them, when **HEADER** is used).

fusion matrices, the performance at each step of the cascade.

1) F_0 step: Figure 7 shows the 3-by-3 confusion matrices obtained at step F_0 with different feature configurations.

The last sharing step is unsurprisingly the easiest one to be identified, since the forensic traces related to that platform are still intact, while they tend to vanish or blend as we move backwards along the sharing chain. In fact, all standalone detectors are able to reach high accuracy values by themselves (Figure 7a–c). Nevertheless, one can observe some interesting results. First, **HEADER** features are able to detect the last step $\mathbf{C}[0]$ with deterministic accuracy; in fact, the presence and frequency of JPEG header markers are closely related to the processing carried out by the last sharing platform during upload. Secondly, we can note that BKS preserves the performance of the best classifiers among those included in the fusion: when **DCT** and **META** descriptors are combined, the results are identical to the ones obtained with **META** alone (Figure 7d); when **HEADER** is present in the fusion, instead, perfect detection is preserved (Figure 7e–f). This characteristic of **HEADER** features is key at the first backtracking block of the cascade, as it guarantees to precisely identify the last

step of the chain, thus avoiding the propagation of errors throughout the pipeline.

2) F_{-1} step: Figure 8 shows the 12-by-12 confusion matrices obtained at step F_{-1} with different feature configurations; the red grid in overlay is meant to highlight the sub-squares related to chains having $\mathbf{C}[0]$ in common.

The reconstruction of $\mathbf{C}[-1]$ allows observing additional interesting results on the behavior of the cascaded system. The results obtained with the individual feature sets (Figure 8a-c) show that classification errors mostly occur among sharing chains having the same platform as last step. For instance, by looking at the top-left 4-by-4 square of the DCT confusion matrix (Figure 8a), which is related to the chains with $\mathbf{C}[0] = $ FB, we can observe how confusions are mostly confined within it; the same result appears in the other sub-squares and for all descriptors. This is a direct consequence of the cascaded approach, which is preserving the performance of the first step throughout the rest of the pipeline. In particular, we can re-observe the perfect classification of $\mathbf{C}[0]$ obtained with **HEADER** features: in Figure 8c, 100% of classifications is confined within the three respective 4-by-4 squares; however,

Figure 9(a) C[0] = FB

True label \ Predicted	FB	FB-FB	FB-FB-FB	FL-FB-FB	TW-FB-FB	FL-FB	FB-FL-FB	FL-FL-FB	TW-FL-FB	TW-FB	FB-TW-FB	FL-TW-FB	TW-TW-FB
FB	0.83	0.0	0.08	0.0	0.0	0.01	0.02	0.04	0.0	0.0	0.01	0.01	0.0
FB-FB	0.14	0.0	0.68	0.02	0.02	0.02	0.06	0.02	0.0	0.0	0.02	0.01	0.0
FB-FB-FB	0.05	0.0	0.87	0.02	0.02	0.0	0.01	0.01	0.0	0.0	0.01	0.01	0.0
FL-FB-FB	0.01	0.0	0.08	0.8	0.0	0.01	0.02	0.06	0.02	0.0	0.0	0.0	0.0
TW-FB-FB	0.08	0.0	0.07	0.03	0.76	0.0	0.0	0.0	0.0	0.0	0.05	0.01	0.0
FL-FB	0.07	0.0	0.0	0.01	0.0	0.32	0.19	0.32	0.09	0.0	0.0	0.0	0.0
FB-FL-FB	0.14	0.0	0.0	0.0	0.0	0.16	0.35	0.25	0.09	0.0	0.0	0.0	0.0
FL-FL-FB	0.03	0.0	0.0	0.0	0.0	0.05	0.09	0.79	0.05	0.0	0.0	0.0	0.0
TW-FL-FB	0.07	0.0	0.0	0.01	0.0	0.1	0.03	0.04	0.76	0.0	0.0	0.0	0.0
TW-FB	0.17	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.63	0.0
FB-TW-FB	0.02	0.0	0.26	0.0	0.02	0.0	0.0	0.0	0.0	0.0	0.49	0.2	0.0
FL-TW-FB	0.0	0.0	0.0	0.0	0.0	0.08	0.02	0.25	0.08	0.0	0.06	0.51	0.0
TW-TW-FB	0.17	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.63	0.0

Figure 9(b) C[0] = FL

True label \ Predicted	FL	FB-FL	FB-FB-FL	FL-FB-FL	TW-FB-FL	FL-FL	FB-FL-FL	FL-FL-FL	TW-FL-FL	TW-FL	FB-TW-FL	FL-TW-FL	TW-TW-FL
FL	0.62	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.06	0.0	0.17	0.16
FB-FL	0.0	0.27	0.36	0.15	0.2	0.0	0.0	0.0	0.0	0.0	0.01	0.0	0.0
FB-FB-FL	0.0	0.15	0.46	0.18	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
FL-FB-FL	0.0	0.03	0.03	0.94	0.01	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
TW-FB-FL	0.0	0.01	0.08	0.0	0.91	0.0	0.0	0.0	0.0	0.0	0.01	0.0	0.0
FL-FL	0.0	0.0	0.0	0.0	0.0	0.88	0.0	0.0	0.12	0.0	0.0	0.0	0.0
FB-FL-FL	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
FL-FL-FL	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
TW-FL-FL	0.0	0.0	0.0	0.0	0.0	0.0	0.27	0.0	0.73	0.0	0.0	0.0	0.0
TW-FL	0.11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.47	0.21
FB-TW-FL	0.0	0.02	0.09	0.01	0.88	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
FL-TW-FL	0.05	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.13	0.0	0.72	0.11	0.0
TW-TW-FL	0.11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.49	0.2

Figure 9(c) C[0] = TW

True label \ Predicted	TW	FB-TW	FB-FB-TW	FL-FB-TW	TW-FB-TW	FL-TW	FB-FL-TW	FL-FL-TW	TW-FL-TW	TW-TW	FB-TW-TW	FL-TW-TW	TW-TW-TW
TW	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
FB-TW	0.0	0.3	0.28	0.36	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
FB-FB-TW	0.0	0.26	0.31	0.37	0.07	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
FL-FB-TW	0.0	0.0	0.0	0.95	0.0	0.0	0.05	0.0	0.0	0.0	0.0	0.0	0.0
TW-FB-TW	0.0	0.08	0.02	0.06	0.84	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
FL-TW	0.0	0.0	0.0	0.0	0.0	0.21	0.0	0.04	0.25	0.0	0.0	0.5	0.0
FB-FL-TW	0.0	0.05	0.23	0.36	0.0	0.0	0.35	0.0	0.0	0.0	0.0	0.0	0.0
FL-FL-TW	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.22	0.15	0.0	0.0	0.52	0.0
TW-FL-TW	0.0	0.0	0.0	0.0	0.0	0.13	0.0	0.04	0.82	0.0	0.0	0.0	0.0
TW-TW	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
FB-TW-TW	0.0	0.3	0.28	0.36	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
FL-TW-TW	0.0	0.0	0.0	0.0	0.0	0.21	0.0	0.04	0.25	0.0	0.0	0.5	0.0
TW-TW-TW	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0

(a) $\mathbf{C}[0] = \mathrm{FB}$ (b) $\mathbf{C}[0] = \mathrm{FL}$ (c) $\mathbf{C}[0] = \mathrm{TW}$

Fig. 9. Detection performance of backtracking block F_{-2}, which identifies the last three platforms in the sharing chain, $(\mathbf{C}[-2], \mathbf{C}[-1], \mathbf{C}[0])$ and constitutes the final output of the implemented system $\mathcal{F}$. The test is carried out with the fusion of all three feature descriptors (see Table II for the results with different feature configurations). The overall 39-by-39 confusion matrix is reported by means of the three diagonal blocks related to chains with $\mathbf{C}[0]$ in common, namely Facebook (a), Flickr (b) and Twitter (c); all elements outside the diagonal blocks are empty, meaning that the last sharing step is always detected correctly.

HEADER alone provides a poor classification of $\mathbf{C}[-1]$, specially for chains ending in FB and TW (top-left and bottom-right squares). Nevertheless, the fusion function is able to compensate this problem by exploiting the information from the other classifiers: as we can see in Figure 8f and in Table II, the fusion of all the three descriptors allow reaching an overall accuracy close to 80%, with a rejection rate of only 2%. Also, we observe that most of the errors occurring in fused configurations fall in the bottom-right square, which is related to sharing chains with $\mathbf{C}[0] = \mathrm{TW}$ (more on this in Section IV-C).

3) F_{-2} step: Figure 9 shows one of the 39-by-39 confusion matrices obtained at step F_{-2}; due to the considerable size of the matrices at this step, we only report the one related to the fusion of all three classifiers; also, since the matrix is perfectly empty outside the three main blocks on the diagonal, which contain chains having $\mathbf{C}[0]$ in common, we just report said blocks separately, in Figure 9a–c.

At the last step of the pipeline, the final classification involves sharing chains of any length up to $L = 3$, which corresponds to $|\Omega_3| = 39$ classes. Despite the high number of classes, the system is able to reach a 55% overall accuracy (random guess is 2.56%) with the fusion of all feature descriptors. However, we also observe a dramatic increase in the rejection rate with respect to the previous steps (see Table II). Moreover, Figure 9c highlights a clear performance drop in the TW-related square, with respect to the other two, suggesting that sharing chains with $\mathbf{C}[0] = \mathrm{TW}$ are somehow more difficult to distinguish. To explain these results, we first analysed the per-class rejection distribution, discovering that 100% of rejections occurred in sharing chains having TW as their last step, which is also in agreement with the performance drop in Figure 9c.

Such initial clues on the difficulties introduced by the presence of Twitter in sharing chains motivated us to conduct a deeper analysis on feature separability, which is the topic of Section IV-C.

4) State-of-the-art comparison: Table III reports a comparison of state-of-the-art methods for the image recycling problem. Solutions in [11], [12] employ histograms of **DCT** coefficients combined with a deep learning approach based on convolutional neural networks (CNNs). In [19] the authors propose a patch-based CNN in two different configurations: the first one receives only **DCT** features in input (P-CNN), while the second one operates a feature fusion of **DCT** coefficients and metadata (P-CNN-FF). For the proposed method, we report the results related to the fusion of **DCT**, **META** and **HEADER** features. All results are obtained on the R-SMUD dataset [19] and reported separately for chains of up to one ($\mathbf{C}_0$), two ($\mathbf{C}_{-1}$) and three ($\mathbf{C}_{-2}$) sharing steps.

TABLE III
METHOD COMPARISON FOR IMAGE RECYCLING.

	Accuracy on R-SMUD [19]		
	$\mathbf{C}_0$	$\mathbf{C}_{-1}$	$\mathbf{C}_{-2}$
Method	3 classes	12 classes	39 classes
[11]	0.9370	0.3991	0.1729
[12]	0.9481	0.4518	0.1695
P-CNN [19]	0.8963	0.4324	0.1932
P-CNN-FF [19]	0.9987	0.6591	0.3618
Proposed	**1.0000**	**0.7981**	**0.5465**

C. Separability analysis

To formally study feature separability, we started from the definition of the ratio of intra/extra-class nearest-neighbor distance [32], which is formulated as

$$IER = \frac{\sum_{i=1}^{n} d(\mathbf{x}_i, NN(\mathbf{x}_i) \in y_i)}{\sum_{i=1}^{n} d(\mathbf{x}_i, NN(\mathbf{x}_i) \notin y_i)}, \quad (8)$$

where n is the number of samples in the dataset and $NN(\mathbf{x}_i)$ is the nearest neighbour of a given sample $\mathbf{x}_i$. Note that

(a) Average per-class LSR computed for **DCT**, **META** and **HEADER** features.

(b) Aggregated LSR for FB.

(c) Aggregated LSR for FL.

(d) Aggregated LSR for TW.

Fig. 10. Average Local Set Radius (LSR) for each sharing chain and for each of the three feature sets, namely **DCT**, **META** and **HEADER** (a), and boxplots showing LSR values computed for **DCT** features and aggregated by platform (b–d); each boxplot represents, from left to right, the chains ending with a specific sharing platform, the two possible sets of chains having that same platform in $\mathbf{C}[-1]$, and all the other classes (Ω^*). The boxes represent the lower and upper quartile, with whisker showing the minumum and maximum values, the median represented as a yellow line and the mean as a green triangle. Low LSR values are associated to chains ending in or containing Twitter, which makes them hardly separable in the feature space.

$NN(\mathbf{x}_i)$ can either belong to the same class of $\mathbf{x}_i$ ($NN(\mathbf{x}_i) \in y_i$) or not ($NN(\mathbf{x}_i) \notin y_i$).

This measure of feature separability, however, has limitations related to the shape of the samples distribution; also, in our case, $d(\mathbf{x}_i, NN(\mathbf{x}_i) \notin y_i)$ is frequently equal to zero, meaning that for several samples the nearest *enemy* (sample from a different class) is overlapped with the sample itself. Therefore, we decided to focus on the denominator of (8), which is also known as the Local Set Radius (LSR), i.e., the radius of the hypershpere centered in one sample and tangent to the nearest enemy:

$$LSR = d(\mathbf{x}_i, NN(\mathbf{x}_i) \notin y_i) \qquad (9)$$

In Figure 10a it is possible to observe that the average LSR is closer to zero for chains having Twitter as their last sharing

step (right-most part of the graph), thus suggesting a lower separability of such classes.

To further highlight this, Figure 10b–d reports the LSR values aggregated in subset related to the specific platforms, for the **DCT** features (**META** exhibits the same trend and **HEADER** is rather uninformative as the LSR is equal to zero in the majority of cases). Figure 10d demonstrates how the presence of Twitter in the sharing chain affects the average LSR: in fact, all groups have values lower than Ω^*, which contains all classes that do not include Twitter in $\mathbf{C}[0]$ or $\mathbf{C}[-1]$. Moreover, in Figure 10b–c, it is possible to see how the lowest values of LSR for chains having Facebook or Flicker in $\mathbf{C}[-1]$ do occur when Twitter is in $\mathbf{C}[0]$.

From this analysis it is clear that, while Twitter is perfectly recognizable when occurring as the last sharing step, chains that contain Twitter in $\mathbf{C}[0]$ or $\mathbf{C}[-1]$ are not separable with the employed sets of features. In general, we can state that

TABLE IV
PER-STEP PERFORMANCE OF THE INFORMED CASCADE SYSTEM.

	F_0 3 classes		F_{-1} 9 classes		F_{-2} 21 classes	
Single classifiers	ACC		ACC		ACC	
DCT	0.8634		0.6219		0.5104	
META	0.9296		0.7302		0.6141	
HEADER	**1.0000**		0.8046		0.5623	
Random guess	0.3333		0.1111		0.0476	
Fused classifiers	ACC	REJ	ACC	REJ	ACC	REJ
DCT+META	0.9296	0.0000	0.7496	0.0000	0.6399	0.0056
META+HEADER	**1.0000**	0.0000	0.9011	0.0000	0.7774	0.0000
DCT+META+HEADER	**1.0000**	0.0000	**0.9057**	0.0010	**0.8105**	0.0377

Twitter is particularly disruptive with regard to the forensic traces left by the previous sharing platforms, at least for the set of traces considered in this work.

The detection of Twitter at a generic step $F_{-\ell}$ of the reconstruction pipeline should therefore be regarded as a stopping point, given the unacceptable performance of previous sharing detectors. Accordingly, we designed an *informed* version of the cascade architecture that stops the reconstruction process when it encounters Twitter, as discussed in the following final section.

D. Informed framework

The informed framework only differs from the standard system described in Section II by the introduction of an additional stopping condition.

As formulated in (4), a backtracking block $F_{-\ell}$ interrupts the reconstruction process when it receives from the previous step a chain of length $\ell - 1$, meaning that the end of the chain has been reached. A second stopping condition is introduced by the BKS fusion modules, which may reject input samples that fall outside of the learned distribution.

In the informed framework, we simply modify (4) by introducing an additional condition:

$$F_{-\ell}(\mathbf{x}, \mathbf{C}) = \begin{cases} \mathbf{C} & \text{if } \mathbf{C} \in \Omega_{\ell-1} \\ & \text{or } \mathbf{C}[-(\ell-1)] = \text{TW} \\ f_{-\ell}^{\mathbf{C}}(\mathbf{x}) & \text{otherwise} \end{cases} \quad (10)$$

This way, when $F_{-\ell}$ receives a chain that contains Twitter in $\mathbf{C}[-(\ell-1)]$, i.e., the last detected step, the reconstruction stops. Clearly, this modification results in a reduced number of classifiable sharing chains.

In the specific implementation evaluated in this work (with $L = 3$ backtracking blocks), all chains of the form $(*, *, \text{TW})$ and $(*, \text{TW}, *)$ collapse in the classes TW and $(\text{TW}, *)$, respectively, thus obtaining 21 classes at the output of F_{-2}.

Table IV reports the overall accuracy values and rejection rates for each step of the informed system (note that F_0 is not affected by the modification), while Figure 11 shows the final 21-by-21 confusion matrix in output of the F_{-2} backtracking block of the informed system.

In Figure 11 we can observe how confusions typically occur when the same platform is concatenated multiple

Fig. 11. Detection performance of backtracking block F_{-2} in the informed cascade system, which identifies up to the last three platforms in the sharing chain, $(\mathbf{C}[-2], \mathbf{C}[-1], \mathbf{C}[0])$; test carried out with the fusion of all three feature descriptors (see Table IV for the results with different feature subsets); note that all chains of the form $(*, *, \text{TW})$ and $(*, \text{TW}, *)$ are condensed in TW and $(\text{TW}, *)$, respectively.

times: (FB, FB) gets mistaken for $(\text{FB}, \text{FB}, \text{FB})$; (FL, FB) and (FB, FL) are confused with $(\text{FL}, \text{FL}, \text{FB})$ and $(\text{FB}, \text{FB}, \text{FL})$, respectively; the only notable exception is FL being confused with (TW, FL).

With the informed cascade, accuracy values are significantly higher than with the standard framework, reaching 90% at step F_{-1} and 81% at step F_{-2}, with the fusion of all three descriptors. More importantly, rejection rates are dramatically reduced, especially at step F_{-2}, confirming that most rejections were due to the non-separability of Twitter-related sub-chains.

V. CONCLUSIONS

The possibility to reverse engineer the history of a digital content in terms of sharing operations can represent a valuable resource in tracing perpetrators of deceptive visual contents, thus playing a significant role in preserving the trustworthiness of digital media and countering misinformation effects. In this work we addressed the media recycling scenario, with the purpose of reconstructing the sharing history of images through multiple uploads on different social media platforms. We proposed a framework that allows the fusion of heterogeneous feature descriptors, combining them in a cascaded classification system that identifies one step of the sharing chain at a time. Also, we introduced a novel set of container-based features extracted from the header of the image file. Experiments demonstrated that the combination of content- and container-based features outperforms the single classifiers in the identification of the various sharing steps. As

a side result, we observed by means of a feature separability analysis that uploads on Twitter turn out to be particularly disruptive towards the traces left by previous platforms, at least for the considered forensic features, thus hampering the reconstruction of the sharing chain. Taking this into account, and therefore interrupting the process at the first detection of an upload on Twitter, the reconstruction achieves an overall 81% accuracy for chains of up to three steps. Given the flexibility of the proposed method, future works may extend the presented architecture with additional platforms, allowing to reconstruct more complex and diversified sharing chains.

Acknowledgment

References

[1] K. Smith, "126 amazing social media statistics and facts," https://www.brandwatch.com/blog/amazing-social-media-statistics-and-facts/, 2019.

[2] Statista Research Department, "Hours of video uploaded to youtube every minute as of may 2019," https://www.statista.com/statistics/259477/hours-of-video-uploaded-to-youtube-every-minute/, 2021.

[3] J. Thies, M. Zollhofer, M. Stamminger, C. Theobalt, and M. Niessner, "Face2face: Real-time face capture and reenactment of rgb videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[4] M. Ngo, S. Karaoglu, and T. Gevers, "Self-supervised face image manipulation by conditioning gan on face decomposition," *IEEE Transactions on Multimedia*, pp. 1–1, 2021.

[5] T. C. G. Allen, "Artificial intelligence and national security," *Belfer Center Study*, 2017.

[6] L. Verdoliva, "Media forensics and deepfakes: an overview," *IEEE Journal of Selected Topics in Signal Processing, in press*, 2020.

[7] C. Pasquini, I. Amerini, and G. Boato, "Media forensics on social media platforms: a survey," *EURASIP Journal on Information Security*, vol. 2021, no. 1, pp. 1–19, 2021.

[8] M. Cheung, J. She, and Z. Jie, "Connection discovery using big data of user-shared images in social media," *IEEE Transactions on Multimedia*, vol. 17, no. 9, pp. 1417–1428, 2015.

[9] B. E. Mada, M. Bagaa, and T. Taleb, "Trust-based video management framework for social multimedia networks," *IEEE Transactions on Multimedia*, vol. 21, no. 3, pp. 603–616, 2019.

[10] M. Moltisanti, A. Paratore, S. Battiato, and L. Saravo, "Image manipulation on facebook for forensics evidence," in *International Conference on Image Analysis and Processing*, 2015, pp. 506–517.

[11] R. Caldelli, R. Becarelli, and I. Amerini, "Image origin classification based on social network provenance," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 6, pp. 1299–1308, 2017.

[12] I. Amerini, T. Uricchio, and R. Caldelli, "Tracing images back to their social network of origin: A CNN-based approach," in *IEEE Workshop on Information Forensics and Security (WIFS)*, 2017, pp. 1–6.

[13] R. Caldelli, I. Amerini, and C. T. Li, "PRNU-based image classification of origin social network with CNN," in *European Signal Processing Conference (EUSIPCO)*, 2018, pp. 1357–1361.

[14] I. Amerini, C.-T. Li, and R. Caldelli, "Social network identification through image classification with CNN," *IEEE Access*, vol. 7, pp. 35 264–35 273, 2019.

[15] A. Mazumdar, J. Singh, Y. S. Tomar, and P. K. Bora, "Detection of image manipulations using siamese convolutional neural networks," in *Pattern Recognition and Machine Intelligence*, 2019, pp. 226–233.

[16] P. Mullan, C. Riess, and F. Freiling, "Forensic source identification using jpeg image headers: The case of smartphones," *Digital Investigation*, vol. 28, pp. S68 – S76, 2019.

[17] O. Giudice, A. Paratore, M. Moltisanti, and S. Battiato, "A classification engine for image ballistics of social data," in *Image Analysis and Processing - ICIAP 2017*, 2017.

[18] Q. Phan, C. Pasquini, G. Boato, and F. G. B. De Natale, "Identifying image provenance: An analysis of mobile instant messaging apps," in *IEEE International Workshop on Multimedia Signal Processing (MMSP)*, 2018, pp. 1–6.

[19] Q. Phan, G. Boato, R. Caldelli, and I. Amerini, "Tracking multiple image sharing on social networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 8266–8270.

[20] N. Siddiqui, A. Anjum, M. Saleem, and S. Islam, "Social media origin based image tracing using deep CNN," in *2019 Fifth International Conference on Image Information Processing (ICIIP)*, 2019, pp. 97–101.

[21] C. Pasquini, P. Schöttle, R. Böhme, G. Boato, and F. F. Pèrez-Gonzàlez, "Forensics of high quality and nearly identical JPEG image recompression," in *ACM Information Hiding and Multimedia Security Workshop*, Vigo, Galicia, Spain, 2016, pp. 11–21.

[22] T. Pevny and J. Fridrich, "Detection of double-compression in jpeg images for applications in steganography," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 2, pp. 247–258, 2008.

[23] E. Kee, M. K. Johnson, and H. Farid, "Digital image authentication from jpeg headers," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 3, pp. 1066–1075, 2011.

[24] P. Yang, D. Baracchi, M. Iuliani, D. Shullani, R. Ni, Y. Zhao, and A. Piva, "Efficient video integrity analysis through container characterization," *IEEE Journal of Selected Topics in Signal Processing*, vol. 14, no. 5, pp. 947–954, 2020.

[25] P. Harvey, "Exiftool," https://exiftool.org/, 2021.

[26] D. Ruta and B. Gabrys, "An overview of classifier fusion methods," *Computing and Information systems*, vol. 7, no. 1, pp. 1–10, 2000.

[27] F. Moreno-Seco, J. M. Inesta, P. J. P. De León, and L. Micó, "Comparison of classifier fusion methods for classification in pattern recognition tasks," in *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*. Springer, 2006, pp. 705–713.

[28] Y. S. Huang and C. Y. Suen, "The behavior-knowledge space method for combination of multiple classifiers," in *IEEE computer society conference on computer vision and pattern recognition*. Institute of Electrical Engineers Inc (IEEE), 1993, pp. 347–347.

[29] ——, "A method of combining multiple experts for the recognition of unconstrained handwritten numerals," *IEEE transactions on pattern analysis and machine intelligence*, vol. 17, no. 1, pp. 90–94, 1995.

[30] Š. Raudys and F. Roli, "The behavior knowledge space fusion method: Analysis of generalization error and strategies for performance improvement," in *International Workshop on Multiple Classifier Systems*. Springer, 2003, pp. 55–64.

[31] D.-T. Dang-Nguyen, C. Pasquini, V. Conotter, and G. Boato, "Raise: A raw images dataset for digital image forensics," in *Proceedings of the 6th ACM multimedia systems conference*, 2015, pp. 219–224.

[32] T. K. Ho, "A data complexity analysis of comparative advantages of decision forest constructors," *Pattern Analysis & Applications*, vol. 5, no. 2, pp. 102–112, 2002.