Media source similarity hashing (MSSH): A practical method for large-scale media investigations

By:
Samantha Klier, Harald Baier

From the proceedings of
The Digital Forensic Research Conference
DFRWS APAC 2025
Nov 10-12, 2025

Forensic Science International: Digital Investigation 54 (2025) 301977

Contents lists available at ScienceDirect

Forensic Science International: Digital Investigation

journal homepage: www.elsevier.com/locate/fsidi

DFRWS APAC 2025 - Selected Papers from the 5th Annual Digital Forensics Research Conference APAC

Media source similarity hashing (MSSH): A practical method for large-scale media investigations

Samantha Klier *, Harald Baier

Research Institute CODE, Faculty for Computer Science, University of the Bundeswehr Munich, Munich, Germany

ARTICLE INFO

Keywords:
Similarity hash
Approximate matching
Media forensics
Source camera identification

ABSTRACT

Hash functions play a crucial role in digital forensics to mitigate data overload. In addition to traditional cryptographic hash functions, similarity hashes - also known as approximate matching schemes - have emerged as effective tools for identifying media files with similar content. However, despite their relevance in investigative settings, a fast and practical method for identifying files originating from similar sources is still lacking. For example, in Child Sexual Abuse Material (CSAM) investigations, it is critical to distinguish between downloaded and potentially self-produced material. To address this gap, we introduce a Media Source Similarity Hash (MSSH), using JPEG images as a case study. MSSH leverages structural features of media files, converting them efficiently into Similarity Digests using n-gram representations. As such, MSSH constitutes the first syntactic approximate matching scheme. We evaluate the MSSH using our publicly available source code across seven datasets. The method achieves AUC scores exceeding 0.90 for native images — across device-, model-, and brand-level classifications, though the strong devicelevel performance likely reflects limitations in existing datasets rather than generalizable capability — and over 0.85 for samples obtained from social media platforms. Despite its lightweight design, MSSH delivers a performance comparable to that of resourceintensive, established Source Camera Identification (SCI) approaches, and surpasses them on a modern dataset, achieving an AUC of 0.97 compared to their AUCs, which range from 0.74 to 0.94. These results underscore MSSH's effectiveness for media source analysis in digital forensics, while preserving the speed and utility advantages typical of hash-based methods.

1. Introduction

Hash functions are a fundamental concept in computer science with a wide variety of potential applications. From a forensic point of view (Institute, 2018), their utility is particularly evident in the areas of data integrity verification and data aggregation. The latter can be achieved with cryptographic hash functions, which facilitate the identification of exact duplicates with remarkable accuracy and efficiency. However, identifying *similar* data is a crucial objective, too, hence, the community has developed similarity hashes also referred to as approximate matching schemes (Kornblum, 2006; Roussev, 2010; Breitinger et al., 2013, 2014). Unlike cryptographic hash functions, which operate exclusively at the byte level, similarity hashes can also be used at the syntactical or semantic level (Breitinger et al., 2013). In the context of media files, the primary focus in forensics is to find similar depicted content for which perceptual hashes are a well-established approach (Steinebach, 2023) and of utmost importance, e.g. to cope with the flood of Child Sexual Abuse Material (CSAM) cases.

For example, in 2023 alone, the National Center for Missing and Exploited Children (NCMEC) received approximately 36 million reports of CSAM uploads (National Center for Missing and Exploited Children, 2024), placing an immense burden on digital forensic laboratories worldwide. While investigators can rely on cryptographic and perceptual hash functions to detect exact duplicates or visually similar content, they lack effective technical tools for the automatic and rapid differentiation of content origin—such as distinguishing between downloaded and self-produced material. As a result, investigators often resort to searching the Exif metadata of CSAM for camera models that are linked to a suspect (Orozco et al., 2013), a process that is time-consuming, error-prone, and easily thwarted by removed metadata.

Due to constrained forensic resources and despite the ethical implications of potentially overlooking victims, investigators regard such compromises, as the lesser of two evils (Casey et al., 2009). Consequently, enabling the aggregation of media files by their source—the last

* Corresponding author.
E-mail addresses: samantha.klier@unibw.de (S. Klier), harald.baier@unibw.de (H. Baier).

https://doi.org/10.1016/j.fsidi.2025.301977

Available online 3 November 2025

processing pipeline component—seamlessly and at scale, as with hash functions may allow investigators to uncover more cases of ongoing child sexual abuse without requiring additional resources. To this end, we propose our concept of a Media Source Similarity Hash (MSSH).

After we introduce the key concepts of similarity hashing and the related field of Source Camera Identification (SCI) (see Section 2), we contribute:

- The lightweight concept of our MSSH, presented on the example of JPEG files, which is based on the extraction of structural data, represented as a set of *n*-grams (see Section 3).
- A publicly available Python implementation of MSSH.[1]
- An evaluation based on JPEG files from seven publicly available data sets on camera devices, models and brands, as well as social media networks, in which our MSSH scored AUC values exceeding 0.9 in 49 out of 50 evaluations (see Section 4).
- A discussion on the potential for expansion to further formats, the identification granularity and the MSSHs placement among other similarity hashes (see Section 5).

Finally, we conclude our paper in Section 6.

2. Background and related work

After the introduction of the characteristics of similarity hashes and particular examples, we regard the domain of Source Camera Identification which techniques have been extended toward screening applications.

2.1. About similarity hashing

The general principles of similarity or approximate matching have been defined by Breitinger et al. (2014). They propose, that a similarity hash function serves to find similarities between two digital artifacts, by providing a value in the range of [0, 1]. Furthermore, similarity hashes use extracted features, and while "a feature can be any value derived from an artifact", the comparison of two features must yield a binary decision whether it matches or not. These features are embraced in a feature set represented as a similarity digest eligible to be compared to the similarity digest of another digital artifact. This framework is applicable on similarity functions, regardless on the abstraction layer they operate on, hence, whether they operate bytewise, syntactically or semantically. However, syntactical similarity hashes are merely seen as a computationally cheap pre-processing step (Breitinger et al., 2014), of which, to the best of our knowledge, no approach is available.

2.2. Established similarity hash functions for media files

Established similarity hash functions, specifically for media files, operate on the semantic level, e.g. PhotoDNA (Steinebach, 2023). Here, the primary aim is to resemble the human perception of visual content which is contrary to our goal to find the source that generated the media file. In contrast, similarity hashes which are generally applicable to any file, such as TLSH (Oliver et al., 2013), ssdeep (Kornblum, 2006) or sdhash (Roussev, 2010), operate on the byte level. Consequently, they are based on the assumption that the similarity of interest is reflected in byte-level encodings. However, this is not valid in our case, as media files are predominantly made out of compressed visual content.

2.3. Source Camera Identification with Sensor Pattern Noise

The primary focus of SCI is the verification of a physical camera as the producer of an image or video. Here, the Sensor Pattern Noise (SPN) approach (Lukas et al., 2006) which extracts specific noise components from an image that can be attributed to a specific imaging sensor, is the most prominent (Klier and Baier, 2025). Traditionally, the SPN approach is expected to achieve a False Acceptance Rate (FAR) of below $2.4 \cdot 10^{-5}$ and a False Negative Rate (FNR) of less than 0.0238 (Goljan et al., 2009). However, the necessary calculations are computationally expensive and the yielded SPN is hard to compress which poses a challenge for storage. Although, there are approaches available that tackle these issues (Valsesia et al., 2015; Li et al., 2018; Bernacki, 2022; Goljan et al., 2010), the overall usability of these adapted SPN methods remain unsatisfactory.

2.4. Source identification with metadata

Metadata based approaches have been proposed which are characterized by their minimal computational costs. For example, Mullan et al. (2019) considered exclusively the number of Exif fields set per Image File Directory (IFD) in images captured with iPhones and achieved a classification accuracy of 0.62 and 0.80 for the model and the iOS version, respectively. A similar approach (Mullan et al., 2020) achieves median accuracies for brand classification in the order of 90 %. Also, post-processing software was identified with an accuracy between 10 % and 75 %. However, this approach can easily be circumvented, by deleting or not recording a few Exif entries, such as the GPS information.

Otherwise, the approach of Iuliani et al. (2018) uses structural metadata of MP4 files for brand identification and classification. More precisely, the field-value attributes, are used to calculate a likelihood ratio for a specific video file based on a set of known brands. In contrast, the approach of López et al. (2020) also extracts the MP4's tree structure, but more comprehensively and uses the obtained information for clustering by brand, model and social media platform, hence a closed set of known target classes is not needed. In contrast, the structure of JPEG images was examined by Gloe (2012) in 2012 from an observational perspective, focusing on image authenticity.

3. Concept

In this section we introduce our MSSH for JPEG files. MSSH is a syntactical approximate matching scheme based on structural features which are saved in a Feature Set and used to generate a Similarity Digest (SD) which can be compared using one of the two provided functions.

3.1. Scope and layer of operation

The primary objective of the MSSH is to convey the resemblance between two sources of media files. However, the term *source* can refer to different things, e.g. the *source* can be a physical device captured the media file, but also a social media network. Therefore, in the scope of MSSH the *source* is the last processing pipeline component that may have altered the syntactic structure of the file at hand.

Therefore, we propose to use structural information of a media file due to the promising results of the works presented in Section 2.4, as well as the appealing cost-effectiveness. Consequently, our MSSH operates on the syntactical layer, and is, to the best of our knowledge, the only similarity hash of this kind. Furthermore, the computational cost of hash generation exhibits a linear time complexity, contingent on the file size, which represents the most efficient achievable scaling.

However, media files are structured based on their respective format which in turn means that each format needs a dedicated implementation. Due to its dominant position, we concentrate on the JPEG file format and discuss the extension to further formats in Section 5.

3.2. Structure extraction

The structure of a JPEG file (Hamilton, 1992; International Organization for Standardization (ISO) and International Electrotechnical

[1] https://github.com/SamKlier/mssh.

Commission (IEC), 1994) has several segments of which some exhibit further sub-structures, hence, is organized hierarchically, as shown in Fig. 1.

3.2.1. Top-level JPEG structure

Overview

Each of the segments is instantiated by a dedicated marker, of which a totality of 64 are available. Therefore, each JPEG file is required to start with a Start of Image (SOI) marker (FF D8) and to end with the End of Image (EOI) marker (FF D9) (see Fig. 1). In between the remaining 62 markers may appear, however, only few are mandatory, such as the Start of Frame (SOF), Define Quantization Table (DQT) and Define Huffman Table (DHT) (CIPA, 2012; International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC), 1994). Additionally, the order in which they may be saved or their quantity is only partially specified which allows a considerable amount of variation in a particular implementation. This is the individual source property, leveraged for the MSSH.

Image selection

We illustrate the procedure based on five smartphone models from three brands, which were selected from the PrnuModernDevices[2] (Albisani et al., 2021) data set, due to the availability of JPEGs captured in standard and bokeh mode. Consequently, the extent to which a non-standard capturing mode affects the file structure and whether different, yet similar, sources indeed exhibit differentiating file structures can be studied.

Example generation

Therefore, Fig. 2 shows the structures of the examples, as extracted by our implementation, which are returned in order of appearance. Moreover, markers that are stripped by metadata removal[3] are marked in red, which applies to all Application Segment (APP) segments here. Moreover, many cameras save unspecified data beyond the EOI marker, referred to as *slack*.

Ideal case

Although the *P20 pro* and *P20 lite* from Huawei are related models, they save their JPEGs with a discriminable structure, as shown in Fig. 2. Moreover, the structure is independent of the selected capturing mode

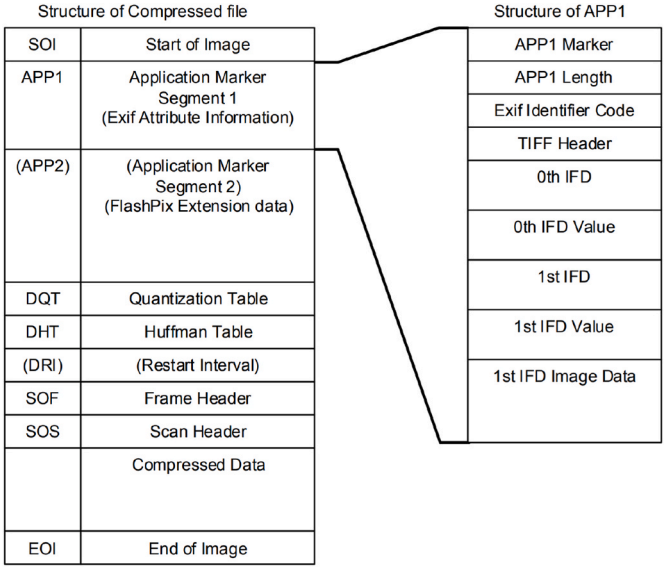

Fig. 1. Basic structure of compressed JPEG files due to CIPA (CIPA, 2012).

Huawei		Samsung			Apple	
P20 pro	P20 lite	S9		S9+	iPhone 11	
std. & bokeh	std. & bokeh	std.	bokeh	std. & bokeh	std.	bokeh
SOI	SOI	SOI	SOI	SOI	SOI	SOI
APP1	APP1	APP1	APP1	APP1	APP1	APP0
APP0	APP0	APP4	APP5	APP4	APP2	APP1
DQT	DQT	APP5	APP4	APP5	APP10	APP1
DQT	DQT	APP11	SOF0	APP11	DQT	APP2
SOF0	SOF0	DHT	DQT	DHT	DRI	APP2
DHT	DHT	DQT	DHT	DQT	SOF0	APP10
DHT	DHT	SOF0	DRI	SOF0	DHT	DQT
DHT	DHT	SOS	SOS	SOS	SOS	DRI
DHT	DHT	EOI	RST0	EOI	RST0	SOF0
DRI	SOS	*slack*	RST1	*slack*	RST1	DHT
SOS	EOI		RST2		RST2	SOS
EOI	*slack*		RST3		RST3	RST0
slack			EOI		RST4	RST1
			slack		RST5	RST2
					RST6	RST3
					RST7	RST4
					[repeat...]	RST5
					EOI	RST6
					slack	RST7
						[repeat...]
						EOI
						slack

Fig. 2. Examples of extracted markers that represent the structures of JPEG files. Markers which are marked red, may be easily removed by a user. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

and even after the deletion of all metadata, which discards all APP structures, the models are differentiable. Interestingly, the *P20 pro* sets the Define Restart Interval (DRI) marker without using the associated Restart Interval Termination (RST) markers which are expected by the specification, hence, may be particularly revealing. In conclusion, this example denotes an ideal outcome for our MSSH.

Capturing modes

In contrast, the closely related but different models, *Galaxy S9* and *S9+* from Samsung, share the same JPEG structure. However, only the *S9* uses a different structure in bokeh mode. Similarly, the structure for the Apple *iPhone 11*, depends on the capturing mode, in particular, two versions of APP1 and APP2, and an APP0 are only available in bokeh mode. Therefore, a one-to-one comparison of media files should only be executed for images captured with the same mode and sources are more discriminable when images from more than one capturing mode are considered. These implications are considered when building the feature set for the MSSH, as proposed in Section 3.3.

RST markers

The amount of available RST markers depends on the number of Minimum Coded Units (MCUs) used by the producer of the JPEG file, as well as on the encoded visual data. Therefore, while RST markers likely provide information about the source, their expected dependency on the visual content is problematic. For now, we leave it by the observation that some models, such as the *iPhone 11* repeat the RST0-RST7 cycle several times whereas other models, such as the *Galaxy S9*, do not.

Summary

Consequently, distinct sources exhibit discriminative structures in their JPEG files, even in the presence of anti-forensic measures, such as the complete removal of common metadata. However, capturing modes

[2] First JPEG (alphabetical order) of each category (flat, nat, bokeh) from *C02*, *C07*, *C14*, *C15* and *C20*.

[3] executed with Exiftool 12.65.

and the visual content depending RST markers have to be considered.

3.2.2. APP1 structure extraction

Overview

The most interesting subordinate structures are the Application Segments (APPs), which are used to save application-specific data, commonly referred to as image metadata. In total, 16 APP markers are available of which some are only used by a few devices or applications (e.g. APP14 by Adobe). Therefore, while the existence of a rare APPs has a high discriminative value on the top-level, they are cumbersome to parse due to proprietary structures. However, the mandatory APP1 has a specified structure, hence, is the only APP considered further.

Interestingly, in bokeh mode, the Apple *iPhone 11* saves two APP1 segments of which the second one saves XMP data, instead of Exif, which is at this point not handled further.

Exif metadata

The APP1 segments, hold Image File Directories (IFDs), as shown in Fig. 3 and contain, among other factors the well-known Exif metadata (CIPA, 2012). Finally, each IFD is further divided into entries of 12 Bytes which respectively start with a marker. In contrast to Mullan et al. (2019) who counted the number of entries per IFD, we consider the markers themselves. Although the *Exif* standard defines some markers, many more are prevalent due to its extensibility. To illustrate, Fig. 4 shows the extracted IFD tags in the order of appearance, for the selected JPEGs (see Section 3.2.1).

Observations

In contrast to the previous results, the extracted Exif markers are independent of the capturing mode. Also, this time the Samsung *S9* and *S9+* exhibit different markers whereas the Huawei models are non distinguishable. Noteworthy, the marker 87 69 is present in all examples and is commonly known as "Maker Notes" which can be seen as an additional non-official IFD. Accordingly, although this may represent the most distinctive Exif information, its use is hindered by proprietary data structures that prevent general parsing; hence, it is excluded from further consideration here.

Therefore, combined with the results from Section 3.2.1, all the models in the considered examples are differentiable based solely on their structural composition.

3.3. Feature set generation

It is essential that the feature set accurately reflects the underlying structure which entails not only the inclusion of the existence of a feature, but also the order or context of its appearance. Therefore, we generate n-grams (Jurafsky and Martin, 2025), which are linked sequences of *n* items, similar to those used in natural language processing. In Fig. 5 we show the mapping of an extracted structure, by the example of 2-g. Due to the fact that every JPEG marker starts with the same Byte, we omit the FF. In contrast, the Exif tags are concatenated, as is. Subsequently, the n-grams are saved in a set, as shown in Listing 1.

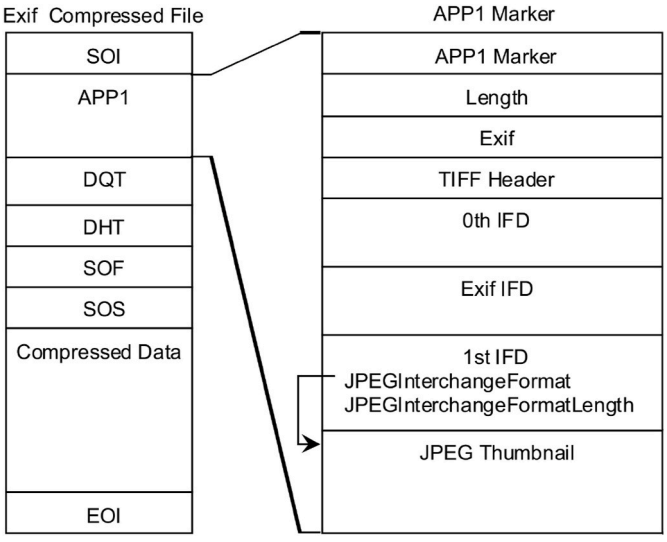

Fig. 3. APP1 structure that holds Exif metadata, as specified by CIPA (CIPA, 2012).

Huawei		Samsung		Apple
P20 pro	P20 lite	S9	S9+	iPhone 11
0100	0100	0100	0112	010F
0101	0101	0101	0213	0110
0102	0102	0112	011A	0112
010F	010F	0213	011B	011A
0110	0110	011A	0128	011B
0112	0112	011B	010F	0128
011A	011A	0128	0110	0131
011B	011B	010F	0131	0132
0128	0128	0110	0132	0213
0131	0131	0131	8769	8769
0132	0132	0132		0006
0213	0213	8769		
8769	8769			
8825	8825			
A40B	A40B			
0180	0180			

Fig. 4. Exif Markers apparent in the example JPEGs.

Listing 1 The Feature Set F for the example in Fig. 5.

```
F = {D8E1, E1E0, E0DB, DBDB, DBC0, C0C4, C4C4, C4DD,
```

Huawei P20 pro						
JPEG Markers			*Exif Tags*			
Marker Name	*Magic*	*2-gram*	*Magic*	*2-gram*		
SOI	FF D8		01 00			
APP1	FF E1	D8 E1	01 01	01 00	01 01	
APP0	FF E0	E1 E0	01 02	01 01	01 02	
DQT	FF DB	E0 DB	01 0F	01 02	01 0F	
DQT	FF DB	DB DB	01 10	01 0F	01 10	
SOF0	FF C0	DB C0	01 12	01 10	01 12	
DHT	FF C4	C0 C4	01 1A	01 12	01 1A	
DHT	FF C4	C4 C4	01 1B	01 1A	01 1B	
DHT	FF C4	C4 C4	01 28	01 1B	01 28	
DHT	FF C4	C4 C4	01 31	01 28	01 31	
DRI	FF DD	C4 DD	01 32	01 31	01 32	
SOS	FF DA	DD DA	02 13	01 32	02 13	
EOI	FF D9	DA D9	87 69	02 13	87 69	
slack	00	D9 00	88 25	87 69	88 25	
			A4 0B	88 25	A4 0B	
			01 80	A4 0B	01 80	

Fig. 5. Creating 2-grams from the JPEG structure of the Huawei P20 pro.

DDDA, DAD9, D900, 01000101, 01010102, 0102010F, 010F0110, 01100112, 0112011A, 011A011B, 011B0128, 01280131, 01310132, 01320213, 02138769, 87698825, 8825A40B, A40B0180}

Consequently, this means that duplicate n-grams are discarded. Arguably, the repetition of an n-gram may hold additional information about the structure. However, some parts of the structure, as the repeated RST markers depend on the content size, as discussed in Section 3.2.1. Instead, longer n-grams (e.g. 3-grams) may model the structure better which is evaluated in Section 4.2. So far, the feature set is based on one media file which enables a one-to-one comparison of JPEGs. However, due to differences in the capturing process (see Section 3.2.1), a feature set that represents a source holistically must be built from several files. Therefore, we construct feature sets for each media file and aggregate them via set union. Consequently, to get a comprehensive view on the source, the selected media files should be diverse, e.g different capturing modes (e.g. bokeh, night mode, action pan) should be used, as well as, different scenes (e.g. macro, landscape) and settings. However, due to the binary nature of sets, there is no value in repeating the same shot.[4] Subsequently, we denote Similarity Digests (SDs) based on such a feature set of several diverse images, as source SD.

3.4. Similarity digest generation

An Similarity Digest (SD) is derived from a Feature Set, obtained from one or multiple JPEG files, by concatenating all n-grams of equal length, sorted alphabetically to improve readability. However, information is not lost by re-ordering, as we operate on sets, as required by the definition of a similarity hash (Breitinger et al., 2014), and the order of the markers is captured due to the use of n-grams.

Consequently, the SD has two parts, which are shown in Listing 2, separated by an empty line. In total, the two parts in the example have a length of 82 Bytes, due to the 164 characters, of which each 2 characters can be represented as 1 Byte by design.

Listing 2 Similarity Digest for the example in Fig. 5.

```
C0C4C4C4C4DDD8E1D900DAD9DBC0DBDBDDDAE0DB E1E0

0100010101010102010201 0F010F011001100112
↪0112011A011A011B011B012801280131013101
↪3201320213021387698769882588 25A40BA40B 0180
```

3.5. Similarity calculation

Finally, to calculate the similarity between two digests, the underlying Feature Sets must be reconstructed; therefore, the chosen n-gram size must be known. To compare the similarity of two sets, the Jaccard Index is commonly used, as shown in Equation (1), where the feature sets F_A and F_B are derived from JPEG files A and B, respectively.

$$J(F_A, F_B) = \frac{|F_A \cap F_B|}{|F_A \cup F_B|} \tag{1}$$

However, the Jaccard Index is symmetrical, hence, makes no difference whether F_A deviates more from F_B or vice versa which is favorable, when two images are compared to each other. But, if a *source SD*, denoted as F_S, was generated based on several diverse JPEGs, the expectation is that the entirety of possible features is represented in the feature set. Consequently, it is expected that F_S contains features which are not present in F_A, even when F_A has been captured by the given source. In contrast, if F_A contains features that are not present in F_S, this is a strong indicator that A was not produced by the considered source. Here, a symmetrical similarity metric is disadvantageous.

Along those lines, the asymmetric Tversky Index (Tversky, 1977) compares a variant to a prototype, as shown in Equation (2). Basically, the Tversky Index introduces weights (α and β) to the Jaccard Index to differentiate on which side a mismatch is prevalent. Consequently, when the weights are both set to 1, the Tversky Index equals the Jaccard Index. Therefore, in our use case, any features that are exclusively apparent in F_S should be tolerated, whereas features exclusively in F_A should be detrimental. Consequently, to measure the similarity between an image and a source, we set $\alpha = 0$ and $\beta = 1$ which is shown in Equation (3).

Both the Jaccard and Tversky indices yield values in the range [0, 1], though a similarity score of 1.0 does not necessarily indicate that the sources are truly identical. Specifically, a Jaccard Index of 1.0 signifies that the two feature sets are identical, while for the Tversky Index used in this work (see Equation (3)), it indicates that F_A contains no features absent from F_S. A comparative evaluation of both indices is presented in Section 4.3.

$$T_{\alpha,\beta}(F_S, F_A) = \frac{|F_S \cap F_A|}{|F_S \cap F_A| + \alpha|F_S \backslash F_A| + \beta|F_A \backslash F_S|} \tag{2}$$

$$T(F_S, F_A) = \frac{|F_S \cap F_A|}{|F_S \cap F_A| + |F_A \backslash F_S|} \tag{3}$$

4. Evaluation

After introducing our data-set, we evaluate our approach two-fold on our publicly available implementation.[5] First we consider the uniqueness of the generated source SDs, depending on n-gram length. Then, we compare the results of the two proposed functions for comparison and their respective classification performance.

4.1. Data sets for evaluation

For the evaluation, our selection criteria on the published image sets are twofold, that is the set provides a ground truth and models represented with more than one device. In all, our search yields seven data sets meeting these requirements and providing in total 87, 739 images, from 287 unique devices of 189 models, as shown in Table 1.

All data sets include, so-called “nat” images which are *natural* shots, including indoor and outdoor scenes captured from various distances. In contrast, “flat” images, depict exclusively a flat and uniformly lit scene, such as a blue sky and are not available for the Forchheim Image Database (FODB) (Hadwiger and Riess, 2021), IMAGIng seNsor idEntification (IMAGINE) (Bernacki and Scherer, 2023) database, as well as, for the Dresden Image Database (DIDB) (Gloe and Böhme, 2010). Although, “flat” images are important for the SPN approach to create a strong reference pattern, they are not introducing the necessary diversity to generate a strong MSSH source SD (see Section 3.3). However, the PrnuMD (Albisani et al., 2021) and the HDR (Al Shaya et al., 2018) data sets also contain images captured in bokeh and High Dynamic Range (HDR) mode, which in turn does allow building a strong source SD.

Additionally, the FODB and VISION (Shullani et al., 2017) provide for each image a variant which was post processed with social-media networks. In total, images are available from five social media networks, namely Facebook (FB), WhatsApp (WA), Instagram (IG), Telegram (TG) and Twitter (TW).

[4] Refers to multiple images captured with nearly identical scene content, composition, and camera settings—for example, the flat images used in the referenced datasets.

[5] https://github.com/SamKlier/mssh.

Table 1
ALL consists of the JPEG images of every other data set.

Data Set	Publ.	Mod.	Dev.	JPEGs	Image T.	Social M.
FloreView (Baracchi et al., 2023)	2023	42	45[a]	6,763	flat, nat	
IMAGINE (Bernacki and Scherer, 2023)	2023	53	67	2,465	nat	
PrnuMD (Iuliani et al., 2021)	2021	17	22	550	flat, nat, bokeh	
FODB (Hadwiger and Riess, 2021)	2021	25	27	23,106	nat	FB, WA, IG, TG, TW
HDR (Al Shaya et al., 2018)	2018	21	23	5,415	flat, nat, HDR	
VISION (Shullani et al., 2017)	2017	29	35	34,427	flat, nat	FB, WA
DIDB (Gloe and Böhme, 2010)	2010	24	68	14,713	nat	
ALL		189[b]	287	87,439	flat, nat, bokeh, HDR	FB, WA, IG, TG, TW

[a] Device D22 (iPhone X) has been excluded due to providing only HEIC images.
[b] Unique models across all data sets, hence, is not the sum of the column.

4.2. Source similarity digest

Several source types are considered and evaluated separately, namely devices, models, brands and social media networks. Therefore, to calculate the source SD for a device, one image per available image type (see Section 4.1) is selected[6] and processed, as described in Section 3, respectively, with 2-grams and 3-grams. In contrast, for the model SDs, we select one image per available device and image type. Accordingly, the brands and the social media networks are processed.

However, due to the varying availability of images, the quality of the source SDs is not constant, e.g. six files can be used for the model SD of the *iPhone SE*, but only one for the *iPhone 11 Pro Max*. Particularly, all device SDs of IMAGINE, FODB and DIDB are based on only one image, hence, are maximally weak. In contrast, the source SDs of each social media network is strong, as they are based on at least 27 images from distinct devices. Furthermore, the brand SDs of Samsung, Huawei, Apple and Xiaomi are also strong, as they are based on 35 images or more from different devices in the ALL data set.

First, the uniqueness of the SDs among the *ALL* data set with regards to the n-gram size is evaluated, as shown in Table 2. Accordingly, increasing the n-gram size from two to three increases the mean SD length significantly by $40 - 80$ %, but, the number of unique SDs per source type remains unchanged. Consequently, 3-grams will not be considered any further. However, the differentiation of devices and models is suboptimal with 49.5 % and 69.3 %, respectively.

4.3. Similarity functions

Both proposed similarity functions are evaluated based on their Area Under Curve (AUC) of the respective Receiver Operating Curves (ROCs) (True Positive Rate and False Positive Rate) per data set and source type, as shown in Table 3. Here, the left side of the table presents the outcomes conducted on the JPEG and APP1 structure. In contrast, the right side illustrates the results of the JPEG structure alone, hence, disregarding any conventional metadata, e.g. due to anti-forensic measures.

Therefore, the proposed Tversky index (S_T) outperforms the Jaccard index (S_J), when a strong source SDs can be calculated, such as for diverse data sets (e.g. PrnuMD) or social media networks and brands. In contrast, the Jaccard index (S_J) is superior when little diversity is available, as for the FODB and IMAGINE datsets or when the APP1

Table 2
Evaluation of source SDs in terms of uniqueness and length with respect to the selected *n*-gram size.

Source	*n*-gram size	mean SD length (B)	n/o SDs	unique SDs
Device	2	67.0 B	287	142/49.5 %
	3	94.0 B	287	142/49.5 %
Model	2	69.2 B	189	131/69.3 %
	3	97.9 B	189	131/69.3 %
Brand	2	105.9 B	32	32/100.0 %
	3	170.2 B	32	32/100.0 %
Social M.	2	28.4 B	5	5/100.0 %
	3	51.6 B	5	5/100.0 %

Table 3
Evaluation of MSSH in terms of AUC, using the two proposed similarity functions, across classification levels: individual devices, models, brands, and social media platforms. Similarity Digests are derived either from both JPEG and APP1 structures, or from the JPEG structure alone. The best result for each category is underlined.

JPEG & APP1			JPEG	
S_J	S_T		S_J	S_T
DEVICE		**Data Set**	**DEVICE**	
0.9906	0.9918	**ALL**	0.9745	0.9684
0.9846	0.9849	**FloreView**	0.9590	0.9546
0.9737	0.9719	**IMAGINE**	0.9457	0.9386
0.9846	0.9838	**FODB**	0.9827	0.9729
0.9173	0.9426	**PrnuMD**	0.8976	0.9263
0.9762	0.9837	**HDR**	0.9693	0.9688
0.9682	0.9642	**VISION**	0.9246	0.9187
0.9687	0.9914	**DIDB**	0.9237	0.9037
MODEL		**Data Set**	**MODEL**	
0.9889	0.9906	**ALL**	0.9739	0.9689
0.9726	0.9720	**FloreView**	0.9576	0.9516
0.9766	0.9753	**IMAGINE**	0.9482	0.9421
0.9873	0.9864	**FODB**	0.9859	0.9755
0.9365	0.9562	**PrnuMD**	0.9239	0.9420
0.9781	0.9862	**HDR**	0.9723	0.9689
0.9736	0.9694	**VISION**	0.9410	0.9327
0.9860	0.9859	**DIDB**	0.9381	0.9236
BRAND		**Data Set**	**BRAND**	
0.7595	0.9893	**ALL**	0.7593	0.9306
0.9652	0.9935	**FloreView**	0.9023	0.9447
0.9386	0.9986	**IMAGINE**	0.8482	0.9607
0.8998	0.9868	**FODB**	0.8347	0.9615
0.9920	0.9943	**PrnuMD**	0.9543	0.9773
0.9616	0.9983	**HDR**	0.8783	0.9486
0.9365	0.9956	**VISION**	0.8485	0.9457
0.9656	0.9866	**DIDB**	0.9255	0.9073
SOCIAL MEDIA		**Data Set**	**SOCIAL MEDIA**	
		FODB	0.7816	0.8505
		VISION	1.0000	1.0000

[6] Files selected for SD calculation are excluded from the evaluation.

Table 4
Comparison of AUC values of MSSH and previously reported approaches, evaluated at the individual device level. Best result per data set is underlined.

	IMAGINE	VISION	DIDB subset	DIDB
Reported by Bernacki (Bernacki, 2022)				
Bernacki (Bernacki, 2022)	0.94	–	0.93	–
Valsesia et al. (Valsesia et al., 2015)	0.78	–	0.81	–
Li et al. (Li et al., 2018)	0.84	–	0.84	–
Lukas et al. (Lukas et al., 2006)	0.89	–	0.90	–
Bondi et al. (Bondi et al., 2016)	0.89	–	0.88	–
Tuama et al. (Tuama et al., 2016)	0.88	–	0.86	–
Mandelli et al. (Mandelli et al., 2020)	0.83	–	0.87	–
Kirchner and Johnson (Kirchner and Johnson, 2019)	0.74	–	0.75	–
Reported by Shullani et al. (Shullani et al., 2017)				
Goljan et al. (Goljan et al., 2009)	–	0.99	–	–
Own Experiments				
Proposed	0.97	0.97	–	0.99

segment is not available. Therefore, the two similarity indices behave in accordance with the proposed hypothesis. However, when the Jaccard Index is superior the difference to the Tversky index is consistently imperceptible, but not vice-a-versa (e.g. Brand classification of ALL). Therefore, in general the used Tversky Index (see Equation (3)) is the better choice.

4.4. Classification performance

4.4.1. Devices and models

The classification performance of the proposed MSSH is evaluated for individual devices and models using the AUC values presented in Table 3. Across all evaluations, the AUCs for S_T exceed 0.9, indicating excellent performance.

In addition, Table 4 presents device classification results reported for SPN-based approaches on the IMAGINE, VISION, and DIDB datasets for comparison. The classic SPN method by Goljan et al. (2009) achieves slightly better performance on VISION's native images compared to the proposed approach. However, the proposed MSSH outperforms eight established SPN methods—each optimized for efficiency—on the more recent IMAGINE dataset. Furthermore, Bernacki (2022) report results on a subset of the DIDB dataset, which is not reconstructible. Therefore, we report the AUC of the proposed method on the complete DIDB dataset. For a discussion of these results, please refer to Section 5.

4.4.2. Brands and social media

In the case of brands, again, the AUCs for S_T exceeds a value of 0.9 for each experiment which is considered excellent. Additionally, the proposed approach yields an AUC of 0.9935 for the complete FloreView dataset on which the SPN approach yielded values in the range of 0.80 – 0.99, depending on the brand and computational cost, as reported by Baracchi et al. (2023). In contrast, the AUC for Social Media networks is considerably lower with a minimum value of 0.85 due to the fact that the Instagram's source SD is a sub-set of Facebook's source SD on the FODB data set.

5. Discussion

The goal of the MSSH is to provide a hash function that preserves the similarity of a media file's source. While the source is not necessarily a physical device, as is the case in Source Camera Identification, the results remain competitive even at the individual device level (see Section 4.4, Table 4). However, these strong results are likely influenced by the limitations of the datasets, which are not well-suited for evaluating true device-level discrimination. But, this means that the "efficient" SCI approaches—unlike the classical SPN-method (Goljan et al., 2009)—have yet to demonstrate conclusive evidence of their ability to distinguish individual devices.

While initial results are promising, a key limitation lies in the lack of representative real-world data. Existing SCI datasets do not adequately capture long-term device usage, user behaviour, or system updates—factors that may influence MSSH-based analysis more significantly than traditional hardware-focused SCI methods. These influences could potentially be beneficial, but further investigation is required to accurately assess the method's actual differentiation granularity.

Currently, the number of unique SDs is relatively low, underscoring the need to enhance the method's ability to differentiate between sources. Notably, substantial structural information remains unexploited, such as additional APP segments, Maker Notes, embedded images, RST marker spacing, and endian encoding—all of which could improve source discrimination.

However, these results support extending the MSSH concept to other media formats, particularly HEIC and MP4, given their widespread practical relevance. Although structural extraction must be tailored to each format, prior studies (see Section 2.4) suggest that the approach is generalizable to complex media container formats.

Furthermore, according to NIST, syntactic similarity hashes—such as MSSH—are primarily designed as computationally efficient preprocessing tools (see Section 2.1). However, MSSH's performance challenges this classification, as it achieves results comparable to established semantic similarity hashes such as TLSH, ssdeep, and sdhash, which report AUCs ranging from 0.65 to 0.98 (Oliver et al., 2013) in their respective domains.

6. Conclusion and future work

Similarity hashes are a valuable component of the digital forensic toolkit, primarily used to reduce the overwhelming volume of data. Traditionally, these hashes focus on visual content similarity, enabling rapid identification of known media. However, early in an investigation—particularly in Child Sexual Abuse Material cases—investigators often face the critical challenge of determining the source of media files, such as distinguishing downloaded material from potentially self-produced content.

To address this, we introduce the first Media Source Similarity Hash, by the example of JPEG files. Our hash leverages structural features, making it computationally efficient and the first of its kind. The method is adaptable to different kinds of sources and extensible, allowing the incorporation of additional structural data. As a result, MSSH enables investigators to assess the source of a media file with the same high usability that conventional hash functions are known for. Notably, MSSH achieves a AUC classification performance that is comparable to, and often exceeds, established Source Camera Identification methods, while avoiding their substantial resource demands.

Nonetheless, further research is needed to assess the method's reliability and differentiation granularity on real-world image sets, along

with a detailed evaluation of its performance. Additionally, future work should investigate the applicability of MSSH to other file formats and examine whether incorporating additional structural features can further improve differentiation granularity.

References

Al Shaya, O., Yang, P., Ni, R., Zhao, Y., Piva, A., 2018. A new dataset for source identification of high dynamic range images. Sensors 18, 3801.

Albisani, C., Iuliani, M., Piva, A., 2021. Checking PRNU usability on modern devices. In: ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, pp. 2535–2539.

Baracchi, D., Shullani, D., Iuliani, M., Piva, A., 2023. FloreView: an image and video dataset for forensic analysis. IEEE Access vol. 11, 109267–109282. https://doi.org/10.1109/ACCESS.2023.3321991.

Bernacki, J., 2022. Digital camera identification by fingerprint's compact representation. Multimed. Tool. Appl. 81, 21641–21674.

Bernacki, J., Scherer, R., 2023. IMAGINE dataset: digital camera identification image benchmarking dataset. In: SECRYPT, pp. 799–804.

Bondi, L., Baroffio, L., Güera, D., Bestagini, P., Delp, E.J., Tubaro, S., 2016. First steps toward camera model identification with convolutional neural networks. IEEE Signal Process. Lett. 24, 259–263.

Breitinger, F., Guttman, B., McCarrin, M., Roussev, V., White, D., 2014. Approximate Matching: Definition and Terminology. US Department of Commerce, National Institute of Standards and Technology.

Breitinger, F., Stivaktakis, G., Baier, H., 2013. Frash: a framework to test algorithms of similarity hashing. Digit. Invest. 10, S50–S58.

Casey, E., Ferraro, M., Nguyen, L., 2009. Investigation delayed is justice denied: proposals for expediting forensic examinations of digital evidence. J. Forensic Sci. 54, 1353–1364.

CIPA, 2012. Exchangeable Image File Format for Digital Still Cameras: Exif Version 2.3. Standard. Camera & Imaging Products Association.

Gloe, T., 2012. Forensic analysis of ordered data structures on the example of jpeg files. In: 2012 IEEE International Workshop on Information Forensics and Security (WIFS). IEEE, pp. 139–144.

Gloe, T., Böhme, R., 2010. The 'Dresden Image Database' for benchmarking digital image forensics. In: Proceedings of the 2010 ACM Symposium on Applied Computing, pp. 1584–1590.

Goljan, M., Fridrich, J., Filler, T., 2009. Large scale test of sensor fingerprint camera identification. In: Media Forensics and Security. SPIE, pp. 170–181.

Goljan, M., Fridrich, J., Filler, T., 2010. Managing a large database of camera fingerprints. In: Media Forensics and Security II. SPIE, pp. 75–86.

Hadwiger, B., Riess, C., 2021. The forchheim image database for camera identification in the wild. In: Pattern Recognition. ICPR International Workshops and Challenges: Virtual Event, January 10–15, 2021, Proceedings, Part VI. Springer, pp. 500–515.

Hamilton, E., 1992. JPEG File Interchange Format, version 1.02.

Institute, N.F., 2018. Forensic Use of Hash Values and Associated Hash Algorithms. Technical Report. Netherlands Forensic Institute.

International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC), 1994. Information Technology – Digital Compression and Coding of continuous-tone Still Images – Part 1: Requirements and Guidelines.

Iuliani, M., Fontani, M., Piva, A., 2021. A leak in PRNU based source identification—questioning fingerprint uniqueness. IEEE Access 9, 52455–52463.

Iuliani, M., Shullani, D., Fontani, M., Meucci, S., Piva, A., 2018. A video forensic framework for the unsupervised analysis of mp4-like file container. IEEE Trans. Inf. Forensics Secur. 14, 635–645.

Jurafsky, D., Martin, J.H., 2025. Speech and Language Processing: an Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition.

Kirchner, M., Johnson, C., 2019. SPN-CNN: boosting sensor-based source camera attribution with deep learning. In: 2019 IEEE International Workshop on Information Forensics and Security (WIFS). IEEE, pp. 1–6.

Klier, S., Baier, H., 2025. Source camera identification-do we have a gold standard? Forensic Sci. Int.: Digit. Invest. 52, 301858.

Kornblum, J., 2006. Identifying almost identical files using context triggered piecewise hashing. Digit. Invest. 3, 91–97.

Li, R., Li, C.T., Guan, Y., 2018. Inference of a compact representation of sensor fingerprint for source camera identification. Pattern Recogn. 74, 556–567.

López, R.R., Luengo, E.A., Orozco, A.L.S., Villalba, L.J.G., 2020. Digital video source identification based on container's structure analysis. IEEE Access 8, 36363–36375.

Lukas, J., Fridrich, J., Goljan, M., 2006. Digital camera identification from sensor pattern noise. IEEE Trans. Inf. Forensics Secur. 1, 205–214.

Mandelli, S., Cozzolino, D., Bestagini, P., Verdoliva, L., Tubaro, S., 2020. CNN-based fast source device identification. IEEE Signal Process. Lett. 27, 1285–1289.

Mullan, P., Riess, C., Freiling, F., 2019. Forensic source identification using JPEG image headers: the case of smartphones. Digit. Invest. 28, S68–S76.

Mullan, P., Riess, C., Freiling, F., 2020. Towards open-set forensic source grouping on JPEG header information. Forensic Sci. Int.: Digit. Invest. 32, 300916.

National Center for Missing & Exploited Children, 2024. 2023 cybertipline reports by electronic service providers (esp). https://www.missingkids.org/content/dam/missingkids/pdfs/2023-reports-by-esp.pdf.

Oliver, J., Cheng, C., Chen, Y., 2013. TLSH–a locality sensitive hash. In: 2013 Fourth Cybercrime and Trustworthy Computing Workshop. IEEE, pp. 7–13.

Orozco, A.S., González, D.A., Corripio, J.R., Villalba, L.G., Hernandez-Castro, J., 2013. Techniques for source camera identification. In: Proceedings of the 6th International Conference on Information Technology, pp. 1–9.

Roussev, V., 2010. Data fingerprinting with similarity digests. In: Advances in Digital Forensics VI: Sixth IFIP WG 11.9 International Conference on Digital Forensics, Hong Kong, China, January 4-6, 2010, Revised Selected Papers 6. Springer, pp. 207–226.

Shullani, D., Fontani, M., Iuliani, M., Shaya, O.A., Piva, A., 2017. Vision: a video and image dataset for source identification. EURASIP J. Inf. Secur. 2017, 1–16.

Steinebach, M., 2023. An analysis of PhotoDNA. In: Proceedings of the 18th International Conference on Availability, Reliability and Security, pp. 1–8.

Tuama, A., Comby, F., Chaumont, M., 2016. Camera model identification with the use of deep convolutional neural networks. In: 2016 IEEE International Workshop on Information Forensics and Security (WIFS). IEEE, pp. 1–6.

Tversky, A., 1977. Features of similarity. Psychol. Rev. 84, 327.

Valsesia, D., Coluccia, G., Bianchi, T., Magli, E., 2015. Compressed fingerprint matching and camera identification via random projections. IEEE Trans. Inf. Forensics Secur. 10, 1472–1485.