EAI Endorsed Transactions
on Internet of Things
Research Article EAI.EU

Scalable Image Clustering to screen for self-produced CSAM

Samantha Klier* and Harald Baier

Research Institute CODE, Universität der Bundeswehr München, Munich, Germany

Abstract

The number of cases involving Child Sexual Abuse Material (CSAM) has increased dramatically in recent years, resulting in significant backlogs. To protect children in the suspect's sphere of influence, immediate identification of self-produced CSAM among acquired CSAM is paramount. Currently, investigators often rely on an approach based on a simple metadata search. However, this approach faces scalability limitations for large cases and is ineffective against anti-forensic measures. Therefore, to address these problems, we bridge the gap between digital forensics and state-of-the-art data science clustering approaches. Our approach enables clustering of more than 130,000 images, which is eight times larger than previous achievements, using commodity hardware and within an hour with the ability to scale even further. In addition, we evaluate the effectiveness of our approach on seven publicly available forensic image databases, taking into account factors such as anti-forensic measures and social media post-processing. Our results show an excellent median clustering-precision (*Homogeinity*) of 0.92 on native images and a median clustering-recall (*Completeness*) of over 0.92 for each test set. Importantly, we provide full reproducibility using only publicly available algorithms, implementations, and image databases.

Received on 22 December 2023; accepted on 03 July 2024; published on 15 July 2024

Keywords: CSAM, clustering, metadata, EXIF, digital image forensics, data science, anti-forensic, source camera identification

doi:10.4108/eetiot.6631

1. Introduction

Today, investigators are faced with a growing volume of Child Sexual Abuse Material (CSAM) cases [36] and the data involved often reaches hundreds of thousands of CSAM instances [11, 41]. Although low in frequency, CSAM cases with an initial suspicion of actual sexual child abuse are the highest priority for investigators. In these cases, it is critical to identify evidence of actual sexual child abuse as quickly as possible to protect children from continuing harm. With respect to digital forensics, the detection of such evidence is primarily the identification of self-produced CSAM among acquired CSAM. This is preferably done on the crime scene, as suggested by the Computer Forensics Field Triage Process Model (CFFTPM) introduced by Rogers et al. [42].

In contrast, most CSAM cases originate from automated reports based on hash-known CSAM uploaded to a Electronic Service Provider (ESP), such as the CyberTipline reports. In 2022, more than 1.5 million CyberTipline reports were tracked in the US alone, translates into incomprehensible 4.7 CSAM uploads per 1,000 population [37], flooding digital forensic laboratories. Although these cases have no initial suspicion of actual sexual child abuse, there is a non-negligible overlap between suspects possessing CSAM on the one hand and committing hands-on sexual abuse on the other, as noted by Bissias et al. [7].

Investigators are fully aware of this problem and the demanded triage [11, 22, 42] has become a fact for CSAM cases, due to limited digital forensic resources and despite the ethical considerations of overlooking victims in CSAM cases, as the lesser of two evils, as pointed out by Casey et al. [11]. However, the investigators still lack adequate technical and conceptual support. Therefore, they search for CSAM

*Corresponding author. Email: samantha.klier@unibw.de

captured by a camera model used by the suspect [38], which is time consuming, insufficiently reduces the amount of data, and is futile in the presence of anti-forensic measures.

Contributions and organization of paper With this paper we leverage the investigator's traditional approach and contribute further steps towards an efficient and effective screening of CSAM for self-produced CSAM. Our contributions are as follows.

We show that the inherent value of the traditional approach is that it can be confirmed by any person in court and avoids technical discussions, but, it struggles with large image sets and fails in the presence of anti-forensic measures (Section 2). Therefore, we present our screening concept, which retains the advantages, but can be applied to very large image sets (more than 100,000 images), is resilient to anti-forensic actions, and reduces the required expert time (Section 3).

Our key contribution is the translation of a complex digital forensic problem into a format compatible with advanced data science tools, which is by no means obvious and includes the selection of metadata as the best-suited data input and the development of a custom distance metric tailored to the problem domain (Section 4). Subsequently, we implement a proof of concept based on the open-source tools UMAP and HDBSCAN, specifically selected to suit our use case (Section 5).

This way, we accomplish a significant breakthrough by successfully clustering more than 130,000 images, surpassing the previous limitations of clustering sets of more than 10,000 images, as we show in our evaluations (Section 6). We perform our clustering in less than 17 minutes once metadata is available or in less than 50 minutes including all steps on commodity hardware and, as we show empirically, scales linearly with the number of images, allowing even larger image sets to be clustered. We evaluate the effectiveness of the clustering on the basis of seven publicly available image databases with respect to the source camera and achieve an excellent median Completeness and good Homogeneity of more than 0.95 and 0.71, respectively, even in the presence of anti-forensic measures. Our approach is able to cluster images after post-processing by social networks, based on their software stack, with an excellent median Completeness of 0.97, but with a low Homogeneity, of less than 0.30.

Finally, we set our approach in context to the related work in Section 7 and conclude our paper in Section 8.

2. The traditional procedure

We begin our explanation by looking at the final step in the forensic process, the presentation of the results in court, which demonstrates the value of the traditional approach, before pointing out its problems.

The value of the traditional procedure When a digital forensic investigation is completed, the results are often reported in a form suitable for non-technical target groups in courts [10]. Therefore, every fact presented in the court must be explained in a way that is understandable to an ordinary person. In our use case of correlating CSAM images to actual sexual abuse by a suspect, a sample of the traditional procedure is as follows: suppose digital forensic analysis reveals two images on the suspect's devices that are presented in court. One image clearly shows the suspect (e.g., the image contains his face), who is wearing a T-shirt with flashy pattern and a bruise on the thumb. The second image shows sexual child abuse conducted by a person without revealing the face, but in a T-shirt with the same flashy pattern and with the same bruise on the thumb.

This is how investigators and prosecutors traditionally proceed [25], linking images based on their actual content until a connection to the suspect is established. This brings digital evidence back into the real world and hence avoids technical discussions in court. Consequently, anyone in a court is able to judge whether a suspect can be considered guilty. But before a prosecutor can present such evidence in court, the images that link a crime to a suspect must be found in sets containing hundreds of thousands of images.

The problem of the traditional procedure Eventually, an investigator is interested in evidence for sexual child abuse (ESCA) by the suspect, which is a set of images, which we denote as I_{ESCA} and is a subset of all images of a CSAM case we define as $I = \{i_1, i_2, ..., i_n\}$. We divide the set I into two subsets, I_{CSAM} which contains all instances of CSAM, and I_P which contains all personal images of the suspect, such as vacation images. Some of these personal images identify the suspect, which we denote as I_{ID}. Consequently, the set I_{ESCA} contains CSAM images that are also personal images, as there is some link to images that identify the suspect. In all we consider the sets

$$I_{ESCA} = \{x \in I_P \mid \exists x.x \in I_{CSAM} \wedge \exists x.x \in I_{ID}\}$$
$$I_P \subseteq I \qquad I_{ID} \subseteq I_P$$
$$I_{CSAM} \subseteq I$$

Obviously, the identification of I is a standard task in digital forensics (e.g., due to known magic bytes in the common file type headers). Furthermore, the extraction of its subset I_{CSAM} can be achieved for known CSAM by hash databases (using both cryptographic

and perceptual hashes) and is increasingly supported by artificial intelligence approaches for yet unknown CSAM [25, 39]. But due to their case-specific and vague definition I_P and its subset I_{ID} are not generically extractable and neither is I_{ESCA}. For example, an identifying image may not show the suspect's face but rather moles that can be verified with a physical examination of the suspect.

Therefore, investigators approximate I_P roughly by searching for images that were taken with a camera model known to be used by the suspect based on easily and quickly extracted image metadata, such as the *Make* and *Model* fields of the Exif standard [12, 38, 45]. However, this approach has some major issues: (i) It fails if none of the fields are set, e.g., due to anti-forensic measures in the form ofExifremover tools that are easily available and usable. (ii) Obtaining the knowledge of the cameras used by the suspect is labor intensive and hardly exhaustive. (iii) It still yields too many images for an investigator to detect subtle clues in the content, as this approach is only moderately discriminatory for popular camera models.

3. Machine Learning based screening

Our goal is to substitute the simple metadata search applied by investigators today to address the issues discussed in Section 2, but not to replace the review. Simply put, enable investigators to review potential evidence of actual sexual child abuse first.

Therefore, we propose a two-phase approach, as shown in Figure 1. First, all images must be divided into packages based on a sort criterion, as represented by the boxes; this will be the focus of this paper.

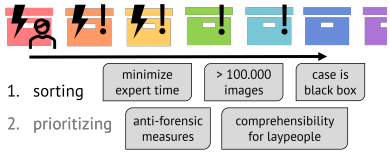

Figure 1. Clusters of case images and boundary conditions of our approach

Next, these packages need to be prioritized for review by an investigator (as indicated by the left-to-right arrow) based on additional information, such as hash-based CSAM detection, which represents known CSAM (black exclamation mark) and AI classification, which represents unknown CSAM (black bolt). To find potentially self-produced CSAM right away, clusters containing unknown CSAM and images identifying the suspect (first package) must first be reviewed. On the contrary, clusters containing hash-known and AI-detected CSAM will be given a lower priority. This may seem counterintuitive at first, but simply finding CSAM is no longer the challenge; our goal is to find evidence of actual sexual child abusefast.

This process must work within the constraints shown at the bottom of Figure 1. Due to time constraints and the limited number of experts available, the time spent on a case by an expert must be minimized. Therefore, we consider a case to be a black box, as any insight into the case must be elaborated. Furthermore, we expect a case to contain at least 100,000 images, and thus the screening must scale well to very large image sets. To make matters worse, we may also be confronted with anti-forensic measures. Consequently, we address each of the three main issues of the traditional approach pointed out in Section 2.

4. Clustering concept

We will now focus on the first step of sorting images into packages, for which we use a clustering approach, as it is superior to classification for the given use case. Subsequently, we turn to formulating our problem in a way that is understood by tools of the data science community, which includes adequate data input and the development of a problem specific distance metric. Finally, we present the metrics on which we will evaluate our approach.

4.1. About clustering

According to Bouveyron et al. [8] the general goal of clustering is to find meaningful groups of data. Typically, the data in these groups will be internally cohesive and separated from one another based on a discriminating property. Hence the purpose is to find pairwise distinct groups whose members have something in common that they do not share with members of other groups. Unlike classification, clustering does not require a training set, is unsupervised, and operates without knowledge on existing classes, which is advantageous for our constraints. However, the disadvantage is that the clusters are not labeled and therefore need to be given meaning in the subsequent prioritization phase.

A clustering function *cluster* in general compares members of the set of all images I and yields a set of clusters C, that is the members of C are depicted as the different packages in Figure 1:

$$C = cluster(I) = \{I_{C_1}, I_{C_2}, ..., I_{C_n}\}$$

For $1 \leq k \leq n$, a cluster is inhabited by $|I_{C_k}|$ images. The data science community offers many clustering approaches as open-source software, e.g. the clustering module of scikit-learn[1]. Although translating a problem from digital forensics to data science is challenging, it enables us to use the most sophisticated tools available

[1] https://scikit-learn.org/stable/modules/clustering.html

for clustering and concentrate mostly on the digital forensics part of our research.

4.2. Adequate data input

First, we need to select the discriminating property for clustering, which in digital forensics is usually the source camera or model. The sensor pattern noise (SPN) (also referred to as Photo Response Non-Uniformity (PRNU) or simply *camera fingerprint*) in different variations is widely used to determine the source camera of an image and has been successfully used for image clustering, for example, by Marra et al. [30] and Lin and Li [27], who clustered nearly 10,000 and 16,000 images, respectively. However, extracting the SPN is computationally expensive, as recently demonstrated by Bernacki [5], who found that the average time to compute the SPN on commodity hardware is in the range of 45-140 seconds, depending on the method. As we aim for a screening approach with an input of more than 100,000 images the computation of the SPNs, which would need approximately 52 days[2], is unbearable.

Therefore, we propose to use image metadata as the cheapest discriminatory feature available. The most commonly known type of metadata for users and investigators alike is Exchangeable Image File Format (EXIF) [12]. EXIF includes the aforementioned *Make* and *Model* fields that directly refer to the capturing device, but also fields pointing to, e.g., a location based on GPS data. However, the metadata saved in images is not completely standardized and is not limited to EXIF [15]. Therefore, the metadata fields that can be extracted from an image set are unknown beforehand. This makes it impossible to select significant fields in advance, a problem also encountered by Mullan et al. [33, 34], who in turn resorted to quantifying the field-value pairs they encountered.

With this approach Mullan et al. [33] achieved a classification accuracy of 0.61 for the identification of iPhone models and a much higher value of 0.80 for iOS versions, respectively, which confirmed their assumption that for smartphones the constantly updating software stack, incl. the operating system and imaging apps interfere with source model identification. However, we aim at finding related images rather than images from the same model, and hence appreciate the impact of the software stack as, e.g. it reflects user habits and time. But, the approach of quantifying the number of set field-value pairs will fail in the presence of anti-forensic measures. Therefore, we propose to extend the approach by taking into account all metadata with their concrete content.

[2] $\frac{45*100,000}{60*60*24} \approx 52,08$

Therefore, we let the extracted metadata of an image $i_j \in I$ be m_{i_j}. We model the metadata element m_{i_j} as a set of field-value pairs

$$m_{i_j} = \{(f_1, v_{1_j}), (f_2, v_{2_j}), ..., (f_l, v_{l_j})\}.$$

Note that the fields in m_{i_j} depend on the fields extracted from the entire image set I. Therefore, the values of the metadata element m_{i_j} are empty if the corresponding field is not set in the image i_j. We denote the actual number of fields set in the image i_j by $|m_{i_j}|$, which is as unknown as the contents of the field-value pairs.

4.3. Distance metric

Fortunately, this level of uncertainty is unproblematic for general-purpose clustering algorithms, as they are designed to be applicable to any kind of data. However, to work, they need a metric that computes the so-called *distance* between two data elements on which the clustering results are based. Many general-purpose distance metrics are available, such as Euclidean or Jaccard, but to be able to incorporate the knowledge we have about our particular use case, we define our own. The two most important factors in our data is the actual content, but also the number of fields set (i.e. $|m_{i_j}|$).

We consider the actual content of the metadata by the *agreement* between two metadata elements as defined in Equation (1) which is the number of identical field-value pairs. Therefore, *agreement*serves to measure the match between the two metadata elements as a non-negative integer.

$$agreement(m_{i_x}, m_{i_y}) = |m_{i_x} \cap m_{i_y}|. \tag{1}$$

However, the maximum possible *agreement* depends on the number of fields actually set (=$|m_{i_j}|$), which can vary significantly between two images, even if they are related. This is true not only if anti-forensic measures have been applied to the images, but also if, for example, the GPS has been disabled while one of the images was taken. Therefore, we normalize the *agreement* with respect to the minimum number of fields set in the metadata elements.This means that if there are only a few fields in a metadata element, the absolute agreement is low, but the relative agreement is actually high. Accordingly, the distance function for our clustering is

$$dist(m_{i_x}, m_{i_y}) = 1 - \frac{agreement(m_{i_x}, m_{i_y})}{\min(|m_{i_x}|, |m_{i_y}|)}. \tag{2}$$

4.4. Evaluation metrics for the clustering

Despite the adherence to our general constraints presented in Section 3, we evaluate the results of

our clustering in terms of efficiency and effectiveness in Section 6. While efficiency is measured in terms of practical runtime (in our setting on commodity hardware), we evaluate effectiveness based on ground truth labels as provided by forensic image databases with respect to the source camera and software stack (source camera, including the image capture app and post-processing, e.g., by Facebook).

Since we are clustering, it makes no sense to use any of the well-known classification metrics, such as accuracy, precision or the F1 score [43]. Therefore, we use evaluation metrics that are well established in the data science and clustering community:

1. ARI: The *Adjusted Rand Index (ARI)* is the standard metric to determine the *accuracy* of a clustering algorithm [18]. The ARI is adjusted for chance and bounded between $[-1, 1]$. A score of 0 indicates the result achieved by a random approach, and a score of 1 implies complete accordance to the ground truth classes.

2. COMP: The *Completeness (COMP)* is a suitable metric for effectiveness in the given use case, as it measures how well a clustering keeps items of the same class together [43], and is thus the clustering counterpart to *recall*. A Completeness of 1 indicates a perfect outcome which means that items with a certain discriminative property are actually assigned to the same cluster. Thus, related images will be reviewed together for high completeness scores. We aim at a Completeness of 0.90 or higher.

3. HOM: The *Homogeneity (HOM)* is a second suitable metric for effectiveness in our use case, as it measures how well a clustering separates items of different classes [43], and thus is the clustering counterpart to the *precision*. A Homogeneity of 1 indicates a perfect outcome which means that every cluster is inhabited by items of one class only. Therefore, the investigator only needs to review related images. On the other hand, a low Homogeneity score means that an investigator needs to review more images than necessary, but evidence is preserved.

4. REJR: The clustering can refuse to assign elements to a cluster; therefore, we compute a *Rejection Rate (REJR)* which is the number of rejected elements divided by all elements. An investigator must review these images to preserve evidence.

In general, we consider the evaluation metrics COMP and HOM (as proposed by Rosenberg and Hirschberg [43]) to be more important than ARI and REJR because, for example, investigators should find images taken by the same individual camera in the same cluster (as measured by COMP) and, if possible, only by that specific camera (as measured by HOM).

5. Implementation

Our proof of concept is based on Python and on open source software of the data science community, extracted metadata and the proposed distance metric *dist* (see Equation (2)). All parts of our implementation are open, and available from our cloud storage[3].

Metadata extraction For our implementation we extract the metadata with ExifTool[4] because it is open source, well known in the digital forensic community, can easily be used in the field, is updated regularly by a strong community and provides machine-processable output in the form of a CSV file. Additionally, it extracts metadata from a plethora of fields, such as Extensible Metadata Platform (XMP), ICC profiles, information about the encoding process, and many more, not just EXIF, as the name suggests. However, any other tool for metadata extraction can be used that yields field-value pairs, but the distance metric, especially the intersection of its *agreement* calculation (see Equation 1), must be implemented appropriately.

Reducing the dimensionality From a large image set, significantly more than 100 unique metadata fields can be extracted, as shown in Table 2, so the problem is highly dimensional. To boost the clustering performance [3], we first reduce the dimensionality of the problem using UMAP [32], an open source tool compatible with the well-known scikit-learn. UMAP takes the high-dimensional input data and generates, based on the custom distance metric *dist* (see Equation (2)), an embedding in a lower dimension. With the *dist* metric we have full control over what being related means to UMAP, but it also means that it is impossible to identify which metadata proved to be the most discriminating.

We implement the *agreement* part of the *dist* metric (see Equation 1) based on numerical and string equality. This means, for example, that a close location as represented in GPS metadata will not be counted as an *agreement* , because the string `'16.682329, 64.781043'` of the `GPS Position` field fails to be string equal to `'16.682339, 64.781258'`, though being very close. The same is true for other fields, as well, be it the position of the thumbnail saved in the header or simply the file size. Therefore, more

[3] https://cloud.digfor.code.unibw-muenchen.de/s/AICSEC_ScalableImageClustering

[4] version 12.54 and execute with the arguments `-rb`,https://exiftool.org/

profound implementations of *agreement* are possible, but they exceptionally increase the execution time.

Otherwise, the results of UMAP are highly influenced by the parameter `n_neighbors` which indicates how many nearest neighbors UMAP should expect. This value is usually tuned to a specific problem in the range from 2 to 100. However, as we aim for an approach that is generic and applicable to any image set without prior knowledge of its structure, we keep this parameter fixed in the center of its range (i.e. a value of 50) throughout our evaluation in Section 6. Additionally, we initialize UMAP with a fixed seed[5] and thus, UMAP yields repeatable results, but this setting reduces the runtime efficiency.

The result of UMAP is an embedding of the given data in two-dimensional space, so we have Cartesian coordinates that we can visualize, as shown in Figure 4, and already reveal clusters perceivable by human perception. Note that we do not draw a coordinate system because the coordinates themselves have no meaning, only the distances between each point have.

Clustering Next, we use the low-dimensional embedding of our data as input for HDBSCAN [9, 31] which finally assigns clusters to our images. The HDBSCAN algorithm is a density-based, hierarchical clustering method, that provides a hierarchy from which a simplified tree of significant clusters can be constructed. We make use of HDBSCAN as it is density-based, which means that clusters can have any form or size. In contrast to the best-known clustering algorithm, k-means, extensive knowledge of the data, such as knowing how many sources are involved, is unnecessary, which is important given the black-box constraint. Additionally, HDBSCAN prefers to reject data from clustering instead of being wrong, which enables investigators to review borderline images separately.

Similarly to UMAP, HDBSCAN has a parameter, i.e. `min_cluster_size`, which highly influences its results and indicates how many images in one cluster are expected. Therefore, we set this value accordingly to `n_neighbors` (i.e. a value of 50), throughout our evaluation in Section 6. In contrast, to UMAP, we use HDBSCAN with its default metric (i.e. Euclidean), as we transformed our problem with our custom distance metric to Cartesian coordinates. Finally, HDBSCAN assigns a label from 0 to $|C| - 1$ to the images, indicating to which cluster a image belongs or -1 if the image was rejected from the clustering.

6. Evaluations

We now discuss the image databases used for our evaluations, which includes their metadata composition.

Table 1. Overview of the image databases used, including year of publication, device types and the number of native, social media post-processed and total images, as well as the number of devices and models available.

database	published	device types	native images	social media	total	devices	models
IMAGINE	not yet	SP, DC, AC, UAV	2,465	0	2,465	66	59
PrnuMD	2021	SP	550	0	550	22	17
FODB	2021	SP	3,851	19,255	23,106	27	25
SOCRatES	2019	SP	9,745	0	9,745	102	58
HDR	2018	SP, TAB	5,415	0	5,415	23	21
VISION	2017	SP, TAB	11,732	22,695	34,427	35	29
DIDB	2010	DC	14,713	0	14,713	68	24
Total			48,471	41,950	90,421	343	

We then evaluate runtime efficiency, source camera clustering, and source software stack clustering, all of which were conducted on a regular laptop[6]. Finally, we sum up the limitations of our approach.

6.1. Image databases for verification

The image test sets for our proof of concept are based on seven publicly available forensic image databases. The ground truth in terms of the source cameras and the social media post-processing are known, respectively. We give a summary of the databases used in Table 1, where the term *native* refers to images that are available as stored by the source camera, and images of the *social media* category have been post-processed by a social media service.

Of course, these databases are not a perfect fit for the intended use case, but evaluating our approach under controlled conditions is a necessary intermediate step before challenging real-case data.

In total, we have 48,471 native images from 343 unique devices and 41,950 images post-processed via social media for our tests available. Each database contains at least two devices of the same model to test if an approach can distinguish devices even if they are of the same model. Finally, we report what types of devices are included. Most databases primarily contain images from smartphones (SP) and digital cameras (DC), but some databases also include images from drones (unmanned aerial vehicles, UAV), action cameras (AC), and tablets (TAB).

To obtain as many images as possible for a large-scale test, the *IMAGINE* database was included, although it is not yet published. However, it has been announced and used by Bernacki et al. [6] and is publicly available. It is also the database with the largest variety of device types.

The PrnuModernDevices (*PrnuMD*) database, proposed by Albisani et al. [2], focuses on images captured in different modes as offered by modern smartphones. Therefore, every device was used to capture images in its native and bokeh mode. The latest smartphone model included is from 2019 (Apple iPhone11) and,

[5] so called `random_state`, set to 42

[6] i7-1165G7 CPU, 32 GB RAM, SSD, Windows 11

as such, is the most recently published one of all databases. The PrnuModernDevices database is the smallest of all and contains only about 25 native images per device.

The subsequent recent database is the Forchheim Image Database (*FODB*), proposed by Hadwiger and Riess [17]. Every incorporated device was used to capture the same scenes under the same conditions. This means that each of the 27 images has the same content and each device contributes the same number of images. Furthermore, each image was post-processed using Facebook, Instagram, WhatsApp, Telegram, and Twitter, respectively.

In contrast, the *SOCRatES* database, proposed by Galdi et al. [13], has been created from submissions of the smartphone owners themselves. Most importantly, this introduces heterogeneity in the data, e.g. due to different habits or software versions. The smartphone owners followed a simple guideline to capture the images that i.a. instructed them to capture 50 images of the blue sky or another uniformly colored surface (so-called flat images) and 50 images of any kind of scene. This means that there are approximately 100 images available for each device, 50 of which were taken under nearly identical conditions and content.

Next the *HDR* database, as proposed by Al Shaya et al. [1], focuses on images captured in High Dynamic Range (HDR) or Standard Dynamic Range (SDR) mode. This database also includes shaky images, and about half of the available images are flat images.

The *VISION* database, proposed by Shullani et al. [44], offers native images (and videos that are not considered for this paper) from portable devices in their native state (incl. flat images), as well as post processed by social media. With a native image to device ratio of 335, VISION offers more native images per device than any other included database.

The Dresden Image Database (*DIDB*) from Gloe and Böhme [16] was published 2010 and is the oldest database that we included and offers only images from digital cameras. The Dresden Image Database (DIDB) is no longer available at the published address, but we were able to obtain a copy (which we provide via our cloud service[7]), though not identical to the original DIDB, as devices and images are missing. With a device-per-model ratio of 2.83, the DIDB contains more devices per model than any other included database.

6.2. Metadata analysis

First off, we show the results of our ExifTool-based metadata extraction from the images of each database in Table 2. The *native* column shows that there are a minimum of 168 (VISION database) and a maximum of 999 (IMAGINE database) that can be extracted from the set of native images in each database. In total, we find 1,304 unique metadata fields across all databases. The most unique fields of native images can be extracted from the database with the most diverse device types, i.e. IMAGINE.

```
"Create", "Modify", "Access", "SourceFile",
"^File", "Directory"
```

Figure 2. Regexes to exclude fields that would leak the database structure.

For the next column *removed EXIF* we consider the number of unique fields that remain after the removal of allExiffields. Unsurprisingly, the removal ofExifinformation reduces the number of extractable metadata fields tremendously, but not completely, as at least 32 (DIDB) and at most 174 (PrnuMD) metadata fields are still available. In total, 296 non-EXIF metadata fields are present across all databases. For example, these fields contain technical metadata (e.g. `ChromaticAdaptation`, `ConnectionSpaceIlluminant`) and are usually not deleted to remain private because they are used to display an image correctly. The most unique non-EXIF fields can be extracted from the database with the newest devices, i.e. PrnuMD.

Table 2. Overview of the number of extracted unique metadata fields by image type. Total refers to the number of *unique* fields over all databases.

database	native	removed Exif	social media
IMAGINE	999	99	-
PrnuMD	288	174	-
FODB	189	74	59
SOCRatES	335	131	-
HDR	195	86	-
VISION	168	46	62
DIDB	395	32	-
Total	1,304	296	69

However, as the last column in Table 2 shows, the worst effect in terms of extractable metadata is due to post-processing by social media applications, which not only removes most of the metadata but may even change it, as observed by the IPTC [19].

Due to the sheer volume, we are not able to understand the meaning of each of the 1,304 fields extracted, but fortunately, this is unnecessary, since we are only interested in the patterns we can detect through our clustering approach. However, for our proof of concept, we need to exclude some metadata fields that would leak the unnatural structure of the databases into our results, which mainly refers to the location of the images on the runtime system, the various file timestamps, and the file names. Therefore, we excluded any field that matches any of the regexes

[7] https://cloud.digfor.code.unibw-muenchen.de/s/DIDB

shown in Figure 2. However, when clustering an image set from a real case, this metadata is valuable and should not be excluded.

6.3. Runtime efficiency

In all, we performed 37 runs containing between 550 and 138,892 images to evaluate the runtime efficiency which must be differentiated between the metadata extraction phase and the clustering phase.

In general, the time needed to extract the metadata scales linearly with the number of images and the size of their headers and depends mostly on the speed of the storage. For a large-scale test, we created a real-life-sized test set of 138,892 images containing every native and post-processed image from each database, plus a copy of every native image with removed Exif information. Extracting the metadata for these 138,892 images took 33 minutes on our runtime system, which translates into a processing speed of about 70 extractions per second.

After the metadata extraction phase, the clustering phase consists mainly of computing the distances between the images. UMAP and HDBSCAN approximate the pairwise distances with a nearest-neighbor approach and therefore avoid computing the distances for all possible $\frac{n\cdot(n-1)}{2}$ pairs. Figure 3 shows the empirical results for each set of tests with respect to the number of images included and the time required for the clustering, as well as a linear trend line ($R^2 = 0.86$).

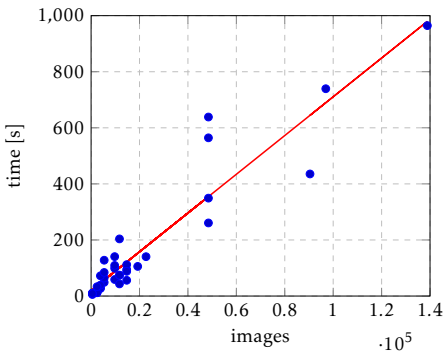

Figure 3. Clustering times of test sets (blue) and linear trend line (red).

Although the scaling complexity of our clustering can be generally described as linearly dependent on the number of images, it also depends on the difficulty of the problem. For example, the four test sets of approximately 50,000 images each take between 260 and 560 seconds, almost a twofold difference, while containing the same number of images. These test sets contain the native images of all databases with different levels of anti-forensic measures (see Section 6.4 for details). However, there is no obvious correlation between the amount of anti-forensic measures applied, the number of metadata fields extracted, the heterogeneity of the data, etc. and the time required for clustering.

In summary, our approach applied to a real case-sized test set requires less than 17 minutes for the clustering phase, along with the 33 minutes for metadata extraction, for a total of 50 minutes for both steps. As a key result in terms of runtime efficiency, our metadata-based approach is able to cluster more than 130,000 images, well over eight times more than any previously proposed approach [21, 24, 26, 27, 30, 40], in less than an hour on commodity hardware, and is expected to scale linearly with the number of images.

6.4. Source Camera Clustering

Because investigators usually look at the source camera of an image to find related images, we evaluate the results of our clustering against the known source cameras of the images.

Table 3. Results of the clustering evaluated to the source camera of an image without anti-forensic measures.

test set	REJR	ARI	HOM	COMP
IMAGINE	0.1927	0.5642	0.7797	0.9957
PrnuMD	0.5800	0.3399	0.4683	1.0000
FODB	0.0104	0.9292	0.9778	0.9789
SOCRatES	0.0923	0.8108	0.9323	0.9769
HDR	0.0669	0.6916	0.9863	0.8221
VISION	0.0934	0.7039	0.9585	0.8420
DIDB	0.0642	0.3055	0.7152	0.6619
ALL	0.0735	0.5790	0.9008	0.8642
median	0.0829	0.6353	0.9166	0.9206

Each test set contains every native image of the corresponding database, while the ALL test set contains the native images of all databases. We first explain the results per database when no anti-forensic measures were applied; the results are shown in Table 3. We then apply anti-forensic actions to an increasing number of images and explain the results based on the median of our metrics, as shown in Table 4.

Results for native images While the results vary considerably from database to database, the Completeness is > 0.8 (except for the obsolete DIDB), the Homogeneity is > 0.7 (except for PrnuMD), and the Rejection Rate is < 0.1 (except for PrnuMD and IMAGINE). In the following, we present an example of an exceptional successful result by discussing the results for the FODB, and then explain the reasons and implications for the weak clustering performance of DIDB, PrnuDB, and IMAGINE.

Figure 4 shows the complete low-dimensional embedding of the FODB. The dots represent the images contained in the FODB, and the colors indicate to which source camera they belong to. Overall, we can see clear and coherent clusters with some errors. For example, images from a Google Nexus 5 (orange) are split into two clusters, while images from the two Huawei P9lite devices (light and dark brown) are clustered together. Interestingly, the two Samsung Galaxy A6 devices (red and yellow) are well separated due to different software

versions. A more quantitative overview of the clusters with respect to their source cameras is shown as a heat map in Figure 5.

The columns show the source cameras, the rows show the clusters, and the last row shows the rejected images. The lighter the color of the box, the more images are assigned (see the scale on the right). We can see that most columns and rows (except the last one) have only one yellow box, which means that most source cameras have all their images assigned to exactly one cluster, which is an almost perfect outcome.

In contrast, we will now illustrate an extraordinarily bad result using the example of DIDB, for which

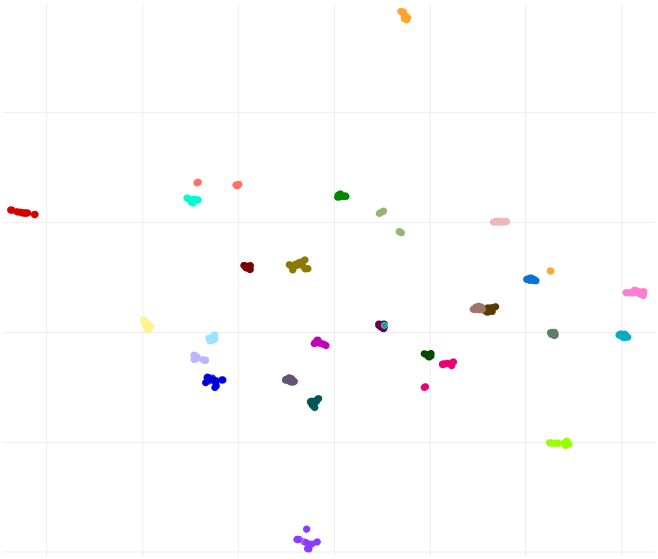

Figure 4. The low-dim. embedding of the FODB. The colors indicate images belonging to a specific source camera.

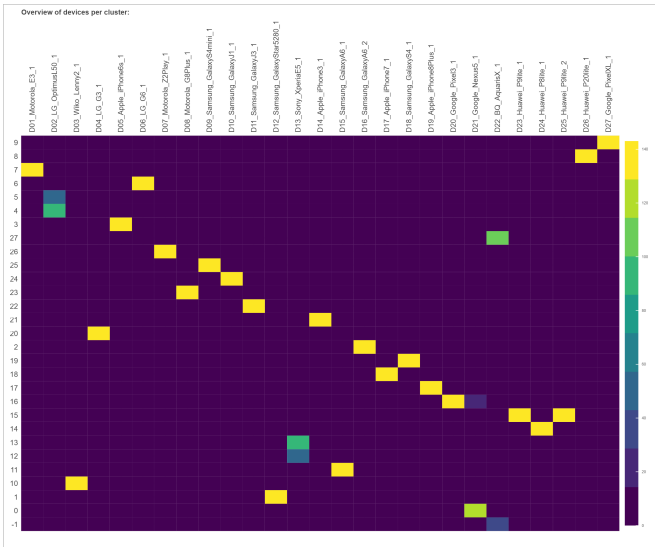

Figure 5. Assignments of images from a specific source camera to a cluster for FODB.

we show a section of its low-dimensional embedding in Figure 6. The figure shows the images from four source cameras of the same Nikon digital camera model. The clustering approach produces two clusters from the low-dimensional embedding and also rejects some images. This effect occurs for several models of the DIDB and is reflected in low ARI, Homogeneity, and Completeness. Although the devices of old digital camera models are indistinguishable by their metadata, a fine-tuning of the used clustering parameters, which are too subtle here, could at least prevent the splitting into two clusters and thus improve the result in terms of Completeness.

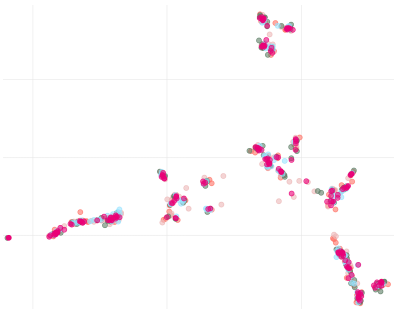

Figure 6. Section of the low-dim. embedding of the DIDB. The colors indicate the images belonging to a specific source camera. Shown here: all four devices of the Nikon CoolPix S710 DC.

On the other hand, the parameters used are too coarse for the PrnuMD and IMAGINE databases, which contain very few images per device; PrnuMD and IMAGINE have a ratio of 25 and 37 images per device, respectively. Thus, these databases score almost perfectly on Completeness, but at the cost of a high rejection rate, low Homogeneity and ARI. Figure 7 shows the heatmap of IMAGINE's clustering, with several boxes per row reflecting the high Completeness but low Homogeinity.

In summary, the median ARI of 0.63 across all test sets is too low for the identification of the source camera. However, our clustering achieves a median Completeness and Homogeneity of 0.92, effectively keeping images from the same source camera together while separating them from others and is therefore sufficiently effective for our sorting phase, as desired.

Results in the presence of anti-forensic measures Using the same test sets, we again evaluate against the source camera of the images, but this time apply anti-forensic measures to 5%, 10%, and 100% of the images in each test set by removing their Exif information. Note that the corresponding native images are not part of the test sets. Table 4 shows the median value of each metric achieved across all test sets and detailed results are shown in the Appendix (see Table I.1).

The median Completeness for test sets without anti-forensic measures is 0.92 and increases with the introduction of anti-forensic measures to 0.99, while the rejection rate decreases. This is counterintuitive at first, but can be explained by Homogeneity and the

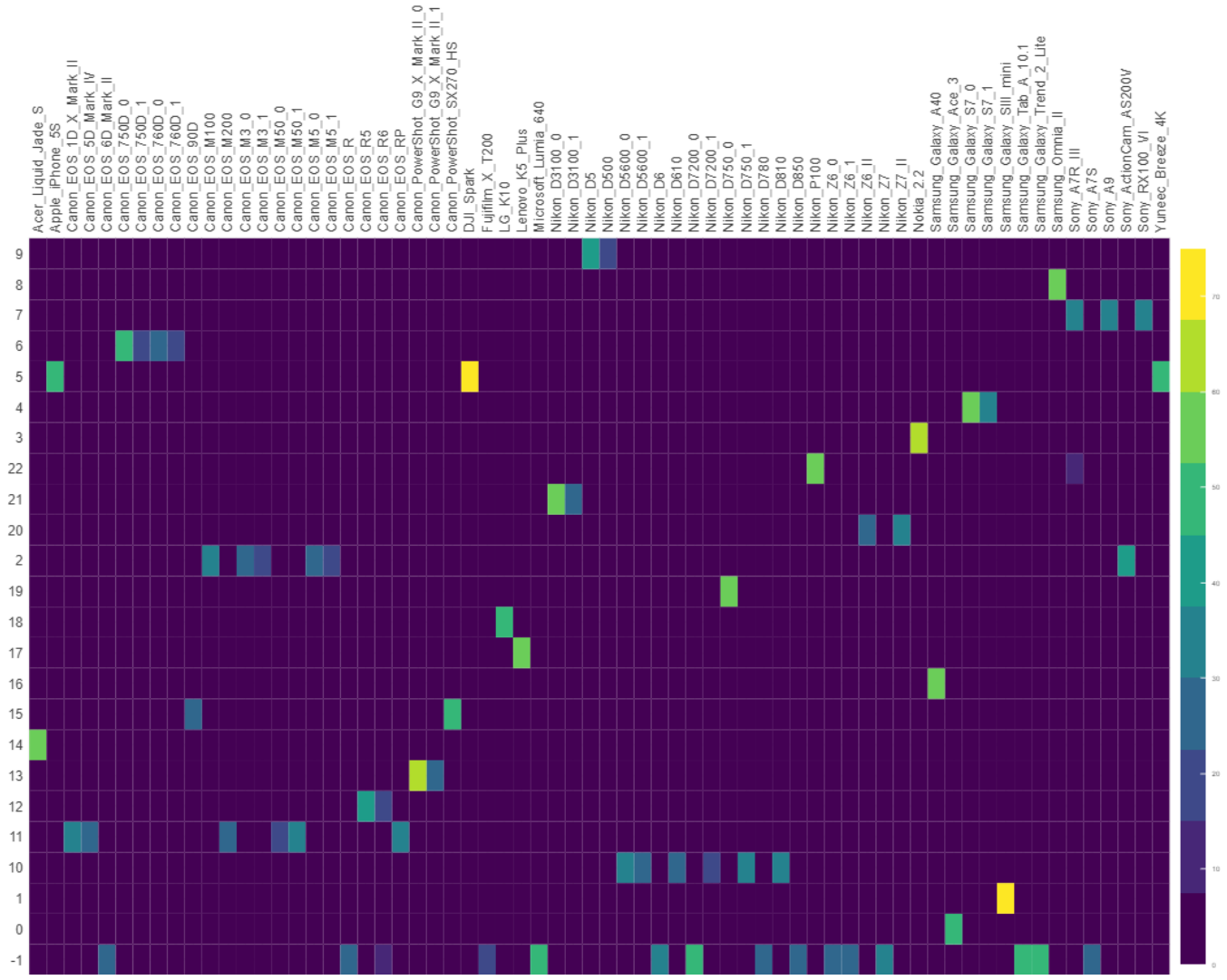

Figure 7. Assignments of images from a specific source camera to a cluster for IMAGINE.

percentage of images with removed Exif	median REJR	median ARI	median HOM	median COMP
0%	0.0829	0.6353	0.9166	0.9206
5%	0.0534	0.3245	0.7179	0.9547
10%	0.0253	0.3246	0.7184	0.9594
100%	0.0102	0.2755	0.7133	0.9909

Table 4. Median metrics achieved across all test sets.

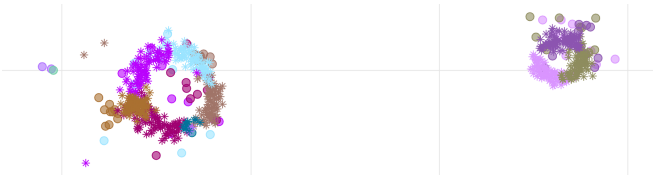

Figure 8. Section of the low-dim. embedding of SOCRatES. Native images are marked by an asterisk, images with removed Exif are marked by a circle.

ARI. The median ARI drops from 0.63 to less than 0.33, while the median Homogeneity drops from 0.92 to less than 0.73. In particular, the achieved ARI is really bad, indicating that the clustering is no longer able to distinguish between the different devices.

However, this means that our clustering will group images from devices that *could* be the source of images with removed Exif information, which is exactly what we want. In Figure 8 we illustrate this effect using two clusters of the SOCRatES test set with 10% of images with removed Exif information as an example. The left cluster contains images from six devices and the right cluster contains images from three devices.

Let us assume that all images with Exif information removed from *Device 173* (*magenta* in Fig. 8) are self-produced CSAM. In this case, knowing that these images are related, an investigator would carefully review all 500 images from the left cluster and would be

presented with every image from *Device 173*, including every image with removed Exif information. This is a significant improvement, as an investigator would normally have to review all 9,745 images or rely on the *Make* and *Model* fields and miss the self-produced CSAM altogether.

6.5. Source Software Stack Clustering

We now evaluate images from FODB and VISION that were post-processed by social media for the source device and the social media type, which is effectively the software stack that generated the image. The results obtained are shown in Table 5.

Table 5. Results of the clustering evaluated by social media type and to the source camera of an image.

social media type	REJR	ARI	HOM	COMP
FODB				
telegram	0.0000	0.0483	0.2779	0.9668
instagram	0.0000	0.0330	0.1839	0.9303
whatsapp	0.0000	0.0572	0.2770	0.9683
twitter	0.0000	0.1401	0.4353	0.9834
facebook[8]	0.0000	0.0288	0.1879	0.9623
median	0.0000	0.0483	0.2770	0.9668
VISION				
facebook high	0.0000	0.0545	0.3089	1.0000
facebook	0.0000	0.2706	0.6178	1.0000
whatsapp	0.0000	0.0543	0.2993	1.0000
median	0.0000	0.0543	0.2779	0.9683

Obviously, the clustering is unable to distinguish the different source devices, as indicated by the extremely low ARI across all test sets, which was expected due to the detrimental effect of social media on metadata (see Table 2). However, the images from a source camera that were post-processed with a specific type of social media are reliably clustered, as indicated by a COMP of above 0.93 across all software stacks.

On the other hand, the medium HOM indicates that the clusters contain images from several source devices that were post-processed with the same social media, which we illustrate with a section of the low-dimensional embedding of the FODB in Figure 9. The large cluster in the lower left corner shows concentrated images of Facebook from different source devices. Interestingly, the images of some devices are distinguishable by our clustering approach (small clusters on the top and right) because some metadata survived the social media post-processing. Therefore, even if the remaining metadata is insufficient to cluster the images by their source devices, they can be successfully clustered by social media type with respect to the source device and, therefore, be reviewed together.

6.6. Summary of Limitations

As discussed in the preceding subsections, there are several limitations to our approach at the moment. Of course, if there is only a limited number of metadata available, the clusters will be undifferentiated, as

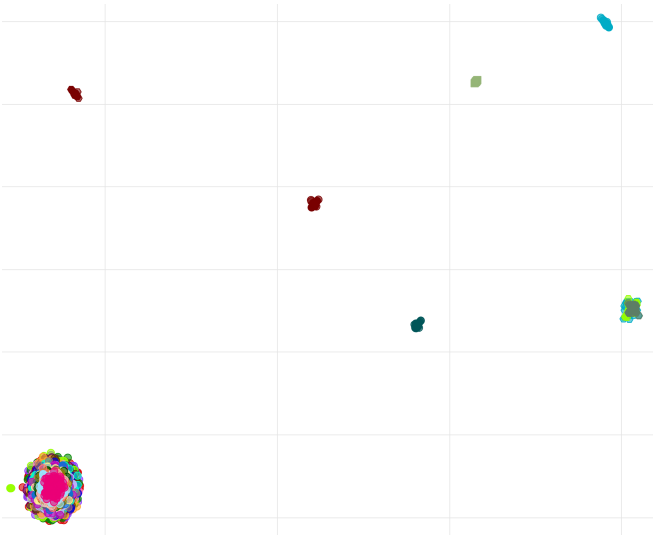

Figure 9. Section of the low-dimensional embedding of FODB's social media images. Colors indicate belonging to a device and the shape the social media type.

observed for the old cameras of the DIDB, for images that have been post-processed by social media or images with removed Exif data. Furthermore, the metadata extraction at the moment is based on the output of ExifTool which may not provide all metadata available in the images, and the used string equality is only a coarse abstraction of the similarity it measures. Despite these limitations, the biggest challenge is the proper determination of the clustering parameters.

7. Related Work

In 2011, image metadata was used by Kee et al. [23] for image authentication and source model identification in traditional digital cameras. Kee et al. focused on the EXIF headers and other technical metadata, such as Huffman coding and quantization tables. They quantified the number of fields set in certain areas of theExifheaders, rather than analyzing the actual data stored. Kee et al. showed that 62% of cameras and 99% of brands had a unique signature in their experiments based on 2.2 million images downloaded from *Flickr*, demonstrating for the first time the efficiency and effectiveness of metadata-based approaches.

This approach was applied to Apple smartphones by Mullan et al. [33], who assumed that the widely varying software stacks of smartphones posed a challenge to identify the source camera based on metadata. Mullan et al. omitted the Huffman coding and used the quantization tables and the signature of theExifheader to show that the approach actually identifies the software stack rather than the smartphone model. Since we are not primarily interested in the exact identification of a specific source device, model,

or brand, we value the identification of the software stack as it is also a meaningful relationship of the images. Mullan et al. achieved an accuracy of 0.82 for the classification of the iOS version and 0.65 for the classification of the smartphone model. Mullan et al. [34] also used this approach to classify the make of a previously unseen camera model.

Unlike these previous works, our approach uses the actual content of the metadata, since counting the number of set fields fails in the presence of anti-forensic measures characterized by metadata deletion. Mullan et al. [34] also point out that image metadata has received little attention so far because it is often claimed to be easily manipulated. In this paper, we show that, while the complete removal ofExifinformation makes the identification of a specific source camera precarious, it still allows the identification of related images.

In addition, Mullan et al. [34] highlights the open set problem, as it is impossible to have a generic, up-to-date, or even case-specific database of devices and models at hand. This is a reality that was also noted by Gloe [14] and picked up by Lorch et al. [28], who proposed an approach to prevent silent classification failures. Since we do clustering instead of classification, we completely avoid this problem, which is also pointed out by Marra et al. [30].

Marra et al. [30] clustered a subset of images from the DIDB based on their SPN. SPN based approaches have been established by Lukas et al. [29] and receive the most attention in the digital forensics community for identifying an image source due to their accuracy. But fingerprint extraction is computationally expensive, as we show in Section 4.2, making it an inappropriate choice for a screening approach. Consequently, Marra et al. clustered a relatively small image set, which totaled 9, 538, and estimated that the scaling of their clustering runtime efficiency is quadratic ($\mathcal{O}(|I|^2)$) which makes the application in a real case impractical.

However, they report an ARI of 0.821 for their largest test set, which is based on the DIDB with 39 devices from ten models, which is significantly higher than the ARI achieved by our metadata-based approach for the DIDB (i.e. 0.3065). Unfortunately, other performance metrics are not available for comparison. Recent studies [2, 4, 20] have shown that SPN-based approaches are generally not suitable for images captured by modern smartphones, as they may be subject to extensive and instantaneous post-processing, such as background blurring, putting doubt on their accuracy today.

8. Conclusion and Future Work

No one knows how much of the CSAM encountered in investigations is self-produced and therefore documents sexual abuse of children by the suspect; we only know how much we find. The digital forensics research community focuses on identifying source devices or models with high accuracy, while sophisticated approaches to screen large data sets are also desperately needed. While there is value in accuracy, it is not the most important metric for a screening approach. Most important is the scalability to enormous data sets while keeping related evidence together, as measured by Completeness.

Therefore, we propose a clustering approach that utilizes cost-effective metadata, as a first step toward truly scalable screening for self-produced CSAM among acquired CSAM. Although this is only a first step, our approach is able to successfully cluster more than 130,000 images in less than an hour on a regular laptop while keeping related images together even in the presence of anti-forensic measures. Thus, our approach is a major improvement over the basic Make/Model-based search used by investigators today.

In our future work, we will address that currently, the quality of the clustering is dependent on the fit of the clustering parameters to the problem. Since our boundary conditions necessitate that a case is a black box, we must tune the parameters at runtime based on internal clustering metrics that do not rely on a known ground truth, such as the Silhouette Coefficient. This implies that the clustering must be repeated multiple times to adjust parameters, which is computationally expensive and may require a surrogate optimization approach [35].

Despite that, after the first successful evaluation of our clustering approach we will target more realistic data sets and also intend to include movies. We are confident that a metadata-based clustering will serve as an ideal foundation for the next step in our screening process, prioritization.

References

[1] Omar Al Shaya, Pengpeng Yang, Rongrong Ni, Yao Zhao, and Alessandro Piva. A new dataset for source identification of high dynamic range images. *Sensors*, 18 (11):3801, 2018.

[2] Chiara Albisani, Massimo Iuliani, and Alessandro Piva. Checking PRNU Usability on Modern Devices. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2535–2539. IEEE, 2021.

[3] Mebarka Allaoui, Mohammed Lamine Kherfi, and Abdelhakim Cheriet. Considerably improving clustering algorithms using umap dimensionality reduction technique: a comparative study. In *Image and Signal Processing: 9th International Conference, ICISP 2020, Marrakesh, Morocco, June 4–6, 2020, Proceedings 9*, pages 317–325. Springer, 2020.

[4] Daniele Baracchi, Massimo Iuliani, Andrea G Nencini, and Alessandro Piva. Facing image source attribution on iphone x. In *Digital Forensics and Watermarking: 19th*

International Workshop, IWDW 2020, Melbourne, VIC, Australia, November 25–27, 2020, Revised Selected Papers 19, pages 196–207. Springer International Publishing, 2021.

[5] Jarosław Bernacki. Digital camera identification by fingerprint's compact representation. *Multimedia Tools and Applications*, pages 1–34, 2022.

[6] Jarosław Bernacki, Kelton AP Costa, and Rafał Scherer. Individual source camera identification with convolutional neural networks. In *Asian Conference on Intelligent Information and Database Systems*, pages 45–55. Springer, 2022.

[7] George Bissias, Brian Levine, Marc Liberatore, Brian Lynn, Juston Moore, Hanna Wallach, and Janis Wolak. Characterization of contact offenders and child exploitation material trafficking on five peer-to-peer networks. *Child abuse & neglect*, 52:185–199, 2016.

[8] Charles Bouveyron, Gilles Celeux, T. Brendan Murphy, and Adrian E. Raftery. *Model-Based Clustering and Classification for Data Science: With Applications in R*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2019. doi:10.1017/9781108644181.002.

[9] Ricardo JGB Campello, Davoud Moulavi, and Jörg Sander. Density-based clustering based on hierarchical density estimates. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 160–172. Springer, 2013.

[10] Eoghan Casey. *Digital Evidence and Computer Crime*. Elsevier, 2011.

[11] Eoghan Casey, Monique Ferraro, and Lam Nguyen. Investigation delayed is justice denied: proposals for expediting forensic examinations of digital evidence. *Journal of forensic sciences*, 54(6):1353–1364, 2009.

[12] CIPA. Exchangeable image file format for digital still cameras: Exif Version 2.32. Standard, Camera & Imaging Products Association, 2019.

[13] Chiara Galdi, Frank Hartung, and Jean-Luc Dugelay. Socrates: A database of realistic data for source camera recognition on smartphones. In *ICPRAM*, pages 648–655, 2019.

[14] Thomas Gloe. Feature-based forensic camera model identification. In Yun Q. Shi and Stefan Katzenbeisser, editors, *Transactions on Data Hiding and Multimedia Security VIII*, pages 42–62, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. ISBN 978-3-642-31971-6.

[15] Thomas Gloe. Forensic analysis of ordered data structures on the example of jpeg files. In *2012 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 139–144. IEEE, 2012.

[16] Thomas Gloe and Rainer Böhme. The'dresden image database'for benchmarking digital image forensics. In *Proceedings of the 2010 ACM symposium on applied computing*, pages 1584–1590, 2010.

[17] Benjamin Hadwiger and Christian Riess. The forchheim image database for camera identification in the wild. In *International Conference on Pattern Recognition*, pages 500–515. Springer, 2021.

[18] Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of classification*, 2:193–218, 1985.

[19] IPTC. Social media sites photo metadata test results, 2020.

[20] Massimo Iuliani, Marco Fontani, and Alessandro Piva. A leak in prnu based source identification—questioning fingerprint uniqueness. *IEEE Access*, 9:52455–52463, 2021.

[21] Xiang Jiang, Shikui Wei, Ting Liu, Ruizhen Zhao, Yao Zhao, and Heng Huang. Blind image clustering for camera source identification via row-sparsity optimization. *IEEE Transactions on Multimedia*, 23:2602–2613, 2020.

[22] Da-Yu Kao, Ni-Chen Wu, and Fuching Tsai. A triage triangle strategy for law enforcement to reduce digital forensic backlogs. In *2020 22nd International Conference on Advanced Communication Technology (ICACT)*, pages 1173–1179. IEEE, 2020.

[23] Eric Kee, Micah K. Johnson, and Hany Farid. Digital image authentication from jpeg headers. *IEEE Transactions on Information Forensics and Security*, 6(3): 1066–1075, 2011. doi:10.1109/TIFS.2011.2128309.

[24] Sahib Khan and Tiziano Bianchi. Fast image clustering based on compressed camera fingerprints. *Signal Processing: Image Communication*, 91:116070, 2021.

[25] Hee-Eun Lee, Tatiana Ermakova, Vasilis Ververis, and Benjamin Fabian. Detecting child sexual abuse material: A comprehensive survey. *Forensic Science International: Digital Investigation*, 34:301022, 2020. ISSN 2666-2817. doi:https://doi.org/10.1016/j.fsidi.2020.301022. URL https://www.sciencedirect.com/science/article/pii/S2666281720301554.

[26] Chang-Tsun Li and Xufeng Lin. A fast source-oriented image clustering method for digital forensics. *EURASIP Journal on Image and Video Processing*, 2017(1):1–16, 2017.

[27] Xufeng Lin and Chang-Tsun Li. Large-scale image clustering based on camera fingerprints. *IEEE Transactions on Information Forensics and Security*, 12(4): 793–808, 2016.

[28] Benedikt Lorch, Franziska Schirrmacher, Anatol Maier, and Christian Riess. Reliable camera model identification using sparse gaussian processes. *IEEE Signal Processing Letters*, 28:912–916, 2021.

[29] Jan Lukas, Jessica Fridrich, and Miroslav Goljan. Digital camera identification from sensor pattern noise. *IEEE Transactions on Information Forensics and Security*, 1(2): 205–214, 2006.

[30] Francesco Marra, Giovanni Poggi, Carlo Sansone, and Luisa Verdoliva. Blind prnu-based image clustering for source identification. *IEEE Transactions on Information Forensics and Security*, 12(9):2197–2211, 2017.

[31] Leland McInnes, John Healy, and Steve Astels. hdbscan: Hierarchical density based clustering. *The Journal of Open Source Software*, 2(11):205, 2017.

[32] Leland McInnes, John Healy, Nathaniel Saul, and Lukas Grossberger. Umap: Uniform manifold approximation and projection. *The Journal of Open Source Software*, 3 (29):861, 2018.

[33] Patrick Mullan, Christian Riess, and Felix Freiling. Forensic source identification using jpeg image headers: The case of smartphones. *Digital Investigation*, 28:S68–S76, 2019.

[34] Patrick Mullan, Christian Riess, and Felix Freiling. Towards open-set forensic source grouping on jpeg header information. *Forensic Science International: Digital Investigation*, 32:300916, 2020.

[35] Juliane Müller. Socemo: surrogate optimization of computationally expensive multiobjective problems. *INFORMS Journal on Computing*, 29(4):581–596, 2017.

[36] National Center for Missing & Exploited Children (NCMEC). 2021 CyberTipline Reports by Country, 2021. https://www.missingkids.org/gethelpnow/cybertipline/cybertiplinedata, last accessed 2023-02-06.

[37] National Center for Missing & Exploited Children (NCMEC). 2022 CyberTipline Reports by Country, 2022. https://www.missingkids.org/content/dam/missingkids/pdfs/2022-reports-by-country.pdf, last accessed 2023-07-13.

[38] AL Sandoval Orozco, DM Arenas González, J Rosales Corripio, LJ Garcıa Villalba, and JC Hernandez-Castro. Techniques for source camera identification. In *Proceedings of the 6th international conference on information technology*, pages 1–9, 2013.

[39] Myeongsuk Pak and Sanghoon Kim. A review of deep learning in image recognition. In *2017 4th International Conference on Computer Applications and Information Processing Technology (CAIPT)*, pages 1–3, 2017. doi:10.1109/CAIPT.2017.8320684.

[40] Quoc-Tin Phan, Giulia Boato, and Francesco GB De Natale. Accurate and scalable image clustering based on sparse representation of camera fingerprint. *IEEE Transactions on Information Forensics and Security*, 14(7): 1902–1916, 2018.

[41] Darren Quick and Kim-Kwang Raymond Choo. Impacts of increasing volume of digital forensic data: A survey and future research challenges. *Digital Investigation*, 11 (4):273–294, 2014.

[42] Marcus K Rogers, James Goldman, Rick Mislan, Timothy Wedge, and Steve Debrota. Computer forensics field triage process model. *Journal of Digital Forensics, Security and Law*, 1(2):2, 2006.

[43] Andrew Rosenberg and Julia Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, pages 410–420, 2007.

[44] Dasara Shullani, Marco Fontani, Massimo Iuliani, Omar Al Shaya, and Alessandro Piva. VISION: a video and image dataset for source identification. *EURASIP Journal on Information Security*, 2017(1):1–16, 2017.

[45] Matthew James Sorrell. Digital camera source identification through jpeg quantisation. In *Multimedia forensics and security*, pages 291–313. IGI Global, 2009.

I. Effectiveness results in presence of anti-forensic measures

In Table I.1 we present our detailed results of the effectiveness of our approach, if anti-forensic activities are used. The discussion of the results is given in Section 6.4.

Test Set	REJR	ARI	HOM	COMP
5% of images with removedExifvalues				
IMAGINE	0.0832	0.3394	0.6920	0.9951
PrnuMD	0.1364	0.3095	0.5052	0.9333
FODB	0.0052	0.5447	0.7892	0.9724
SOCRatES	0.0503	0.3870	0.7640	0.9880
HDR	0.0467	0.6859	0.8902	0.9188
VISION	0.0291	0.2643	0.7087	0.9273
DIDB	0.0646	0.1857	0.5181	0.9448
ALL	0.0565	0.2237	0.7271	0.9645
median	0.0534	0.3245	0.7179	0.9547
10% of images with removedExifvalues				
IMAGINE	0.0592	0.3927	0.7144	0.9990
PrnuMD	0.1000	0.3586	0.5563	0.9428
FODB	0.0254	0.5500	0.8040	0.9622
SOCRatES	0.0000	0.2905	0.7331	0.9909
HDR	0.0124	0.7422	0.9131	0.9348
VISION	0.0165	0.2644	0.6988	0.9329
DIDB	0.0252	0.1669	0.5064	0.9565
ALL	0.0405	0.2221	0.7224	0.9653
median	0.0253	0.3246	0.7184	0.9594
100% of images with removedExifvalues				
IMAGINE	0.0637	0.3632	0.7056	0.9982
PrnuMD	0.1073	0.1555	0.3811	0.9659
FODB	0.0086	0.7092	0.8665	0.9740
SOCRatES	0.0118	0.2972	0.7440	0.9916
HDR	0.0205	0.7457	0.9021	0.9474
VISION	0.0000	0.2537	0.6615	0.9902
DIDB	0.0000	0.1806	0.5071	0.9947
ALL	0.0084	0.2441	0.7210	0.9922
median	0.0102	0.2755	0.7133	0.9909

Table I.1. Results of the clustering evaluated to the source camera of an image with anti-forensic measures applied.