IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 19, 2019, accepted January 6, 2020, date of publication February 5, 2020, date of current version March 2, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2971785

Digital Video Source Identification Based on Container's Structure Analysis

RAQUEL RAMOS LÓPEZ[1,2], ELENA ALMARAZ LUENGO[3], ANA LUCILA SANDOVAL OROZCO[1], AND LUIS JAVIER GARCÍA VILLALBA[1]

[1]Group of Analysis, Security and Systems (GASS), Department of Software Engineering and Artificial Intelligence (DISIA), Faculty of Computer Science and Engineering, Universidad Complutense de Madrid (UCM), 28040 Madrid, Spain
[2]Securitas Direct España, S.A.U., 28224 Madrid, Spain
[3]Group of Analysis, Security and Systems (GASS), Department of Statistics and Operations Research, Faculty of Mathematical Science, Universidad Complutense de Madrid (UCM), 28040 Madrid, Spain

Corresponding author: Raquel Ramos López (raquel.rlopez@securitasdirect.es)

ABSTRACT The mobile device ecosystem has dramatically evolved over the last few years, since users have openly embraced a massive use of mobile phones for different purposes: professional use, personal use, etc. Digital videos can be used to define legal responsibilities or as part of the evidence in trials. The forensic analysis of digital videos becomes very relevant to determine the origin and authenticity of a video in order to link an individual with a device, place or event. The field of forensic analysis of digital videos is constantly facing new and direct challenges. Even though the basic principles of this discipline remain unchanged, numerous issues appear every year that require new procedures and tools. Therefore, it is necessary to provide forensic analysts with techniques to identify the origin of multimedia content. In this paper, the topic of source identification in open scenarios will be discussed, since analysts do not know in advance the set of cameras to which a video belongs so they find it difficult to identify its source. This approach is similar to real-life situations since in most cases, analysts are unaware of the set of video cameras. This paper aims to create a technique that identifies the source of digital videos generated by digital devices through the use of unsupervised algorithms based on the analysis of the structure of multimedia video devices.

INDEX TERMS Acquisition source identification, clustering analysis, container atoms, forensics analysis, video container.

I. INTRODUCTION

The field of mobile devices forensics has emerged in recent years becoming one of the most important areas of investigation, for several reasons. Firstly, the capabilities of smart devices have improved substantially, and now smartphones are being used more than laptops since users have them at their fingertips at any time of the day. Moreover, these smart devices are constantly recording our activities and movements, which in turn give us a vision of our behaviour [1].

The combination of smart mobile phones with social media platforms and cloud storage has allowed video to become a major source of information for many people and institutions. In turn, these digital videos can be made at any time and anywhere for different purposes. They can also be distributed on the internet in a short period of time and sometimes they can show illegal acts such as those related to terrorism, child pornography, industrial espionage, etc. The presence of digital videos in judicial investigations is increasingly common. To address these problems, researchers have developed forensic algorithms that verify the authenticity and source of digital content [2]. Forensic techniques that identify information about the source when multimedia content is generated (images or videos), are divided into two main approaches: on the one hand they serve to verify the origin of a multimedia content and on the other hand, they can detect inconsistencies in the source within the multimedia content that could show signs of a forgery [2], [3]. Numerous investigations develop forensic algorithms to determine the identification of the source of an image but when it comes to digital videos, research is very scarce [3]. In [4] it is suggested that these algorithms utilise traces left by a wide variety of physical and algorithmic components in a camera's processing pipeline. Forensic camera model algorithms have

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

been designed that leverage traces left by demosaicing (Color Filter Array (CFA) method and Demosaicing artefacts), [5]–[7], Joint Photographic Experts Group (JPEG) header information [8]. The majority of existing work has focused on using sensor fingerprints to identify a video's specific source device, for example see [9]–[15].

The analysis of the source of video acquisition is one of the first problems that have emerged in forensic analysis techniques. Inside the identification of the source of acquisition there are two major approaches: closed scenarios and open scenarios. A closed scenario is one in which the identification of the source of the video is made on a set of specific and known cameras. For this approach, a set of videos from each device is normally used to train a classifier and subsequently the source of acquisition of the videos under investigation is predicted.

In [16] it is presented a digital video source identification scheme based on Photo Response Non Uniformity (PRNU) noise and Support Vector Machine (SVM). Given an input video, frames with more significant scene changes are extracted using the colour histogram. A total of 81 functions, which are the Wavelet components of the sensor, are used to train the SVM classifier with training videos. A total of 5 different devices from 5 different brands were used to train the SVM classifier. The results obtained show a success rate of 87% or 90%, depending on the resolution of the video. In an open scenario, the forensic analyst does not initially know the set of devices to which the videos belong to identify their source of acquisition. The objective is not to identify the brand and model of the videos but to be able to group them into disjoint sets in which all their videos belong to the same device. This last approach is more realistic since in many cases the analyst completely ignores the set of devices to which a set of videos can belong. Identifying the device that generates digital content is very important in the context of a judicial process because it can incriminate or delimit responsibilities to a suspect before a criminal act.

This work proposes a technique that identifies the source of acquisition of digital videos generated by different devices. It has been shown that it is possible to identify the source of acquisition of a video through the characteristics of its internal elements and its metadata. These characteristics, which are acquired during its creation process and subsequent processing, are part of what could be considered the DNA of a video, and being analyzed, can show determining information about a digital content. The present investigation focuses on the techniques of identification of source of a video, since it is a little studied field in comparison with the digital images. This paper is divided into 5 sections, the first being this introduction. Section II presents the main developed works in this context. The proposed solutions are presented in Section III. The experiments and their results are presented in Section IV. Finally, in Section V the conclusions drawn from this work are presented.

II. RELATED WORKS

Forensic video analysis techniques still raise many issues to investigate, due to the wide range of possible alterations that can be applied to them. In addition, forensic video analysis has proven to be more difficult with respect to image analysis since the data contained in the videos has higher compression formats than in the case of images.

The video is formed by a sequence of images called frames that vary over time giving a sense of movement. Due to the large volume of information that a video has, it is encoded and decoded using a mathematical algorithm known as a codec. In turn, these already encoded frames are encapsulated along with the audio, metadata and subtitle tracks in a single file known as a multimedia container. In Table 1, it is shown the different elements by which a multimedia container is composed.

TABLE 1. Format video container for .avi, .mp4, .mov, .ogg, .flv, .mkv, etc.

Video codec	Audio codec	Captioning	Metadata
		Video description	
H.264.	AAC	SAMI	MPEG-7
VC-1	WMA	SMIL	CableLabs
Theora	Vorbis	Hi-Caption	TV-ANytime
Dirac 2.1	PCM, etc.	CMML	EBU
H.263, etc.		DFXP	XPM, etc.
		3GPP TS	
		MPSub, etc.	

Multimedia containers or video formats are called computer applications that are capable of storing audio and video, and, in some cases, also subtitles and other additional information.

The most used multimedia containers nowadays are:

- AVI (Audio Video Interleave): A Windows standard multimedia container.
- H264/AVC or MPEG 4 Part 14 (known as .mp4): It is the standardized container for MPEG-4 [17].
- FLV (Flash Video): It is the format used to deliver MPEG video through Flash Player.
- MKV (Mastroska): It is another open specification container that appears in the download of animations.
- MOV: Apple's QuickTime container format [18].
- OGG, OGM, OGV: Open standard containers.

In the most recent literature, it can be find that most of the investigations analyze the internal structure of multimedia containers in the case of the AVI format, the study of MP4, 3GP and MOV containers is almost non-existent.

One of the first works where an analysis of the structures of the videos is made in detail in [19] where AVI and MP4-like (MOV, 3GP, MP4) video streams of mobile phones and digital cameras are analyzed in detail. The authors use customized parsers to extract all file format structures of videos from overall 19 digital camera models, 14 mobile phone models, and 6 video editing toolboxes. They report considerable differences in the choice of container formats, audio and

video compression algorithms, acquisition parameters, and internal file structure. In combination, such characteristics can help to authenticate digital video files in forensic settings by distinguishing between original and post-processed videos, verifying the purported source of a file, or identifying the true acquisition device model or the processing software used for video processing. One of their main findings is that videos from digital cameras and mobile phones often employ different container formats and compression codecs. Mobile phones opt for sophisticated compression algorithms (MP4V, H.26x).

Most digital cameras in their test set prefer a combination of AVI containers and basic MJPEG compression. The structure of AVI and MP4-like containers is not strictly defined. They observed considerable differences both in the order and in the presence of individual data segments. AVI files often contain specific INFO lists or JUNK chunks. MP4-like files may employ various nonstandard Container atoms and different parametrizations of specific atom entries.

In [20] a method for unsupervised analysis of video file containers is introduced, and their authors present two main forensic applications of such method: the first one deals with video integrity verification, based on the dissimilarity between a reference and a query file container; the second one focuses on the identification and classification of the source device brand, based on the analysis of containers structure and content. They tested the effectiveness of both applications on a dataset composed by 578 videos taken with modern smartphones from major brands and models.They conclude that the proposed solution provides an extremely small computational cost as opposed to all available techniques based on the video stream analysis or manual inspection of file containers.

In [21] their authors investigate video content stored in Video Event Data Recorders (VEDRs). VEDRs are used as important evidence when certain events such as vehicle collisions occur. They investigate the video file structure characteristics for each type of video editing software that would leave traces from processing the video editing software. Because such traces are an inherent characteristic of each respective video editing software suite, they can detect the specific video editing software used to manipulate the video, in addition to whether the video was, indeed, manipulated. To evaluate the accuracy of their technique, they examined 296 unmodified Audio Video Interleave (AVI) video files. They performed this study using popular versions of video editing software. As a result, they found that the AVI data structures in modified video files appear consistently according to each video editing software suite. Each resulting data structure is unaffected by the original video file structure.

III. TECHNIQUE DESCRIPTION

The main objective of this work is to propose a technique that allows identifying the source camcorder that generated a digital video. The technique uses clustering algorithms to make the correct grouping of both, brand and model, and the digital video.

To explain the details of the technique, it has been divided into two subsections: first one is explained that it is a multimedia container and which are the best features to perform the identification of the device. In the second subsection, the clustering techniques used to make a correct grouping are described.

A. CONTAINER STRUCTURE ANALYSIS

The elementary structure of a video is the atom. The metadata, video and sound of a video are within them. Container atoms are hierarchical in nature. That is, an atom may contain other atoms, which may contain still others, and so on. The type of atom is specified by a 32-bit unsigned integer, typically interpreted as a four-character ASCII (American Standard Code for Information Interchange) code usually in lowercase letters. It should be noted that there is no rule regarding the Container atoms that must appear and their order, however, most follow a similar structure [22]. This algorithm has been used to extract information from Container atoms. This solution is capable of analyzing multiple information of any video format such as: MP4/H.264, MOV and 3GP video and be able to extract information from Container atoms.

The extraction of Container atoms consists of storing the labels, values and the hierarchy that exists between Container atoms. The process begins by obtaining the initial byte of the atom, size, and the type of the Container atom with a maximum length of 4 bytes formed as a string of characters (eg: fytp, moov, mdat, etc.). Next, the duplicity of Container atoms and the existence of child Container atoms are verified. Finally, a dictionary of a set of Container atoms and tags (Path-tag) with their respective values and orders of appearance is obtained. For a more in-depth study of Container atoms, see [19] and [22]. Table 2 shows the output that is obtained when the atom extraction algorithm is used.

TABLE 2. Example of container atoms extracted from a video.

Path	PathOrder	Field	Value
ftyp	ftyp-1	majorBrand	mp42
ftyp	ftyp-1	minorVersion	1
ftyp	ftyp-1	compatibleBrands	mp41mp42isom
beam	beam-2	byteInitial	28
beam	beam-2	size	42
moov/trak/tkhd	moov-4/trak-2/tkhd-1	version	0
moov/trak/tkhd	moov-4/trak-2/tkhd-1	flags	1
moov/trak/tkhd	moov-4/trak-2/tkhd-1	trackId	1
moov/trak/tkhd	moov-4/trak-2/tkhd-1	trackWidth	48
moov/trak/tkhd	moov-4/trak-2/tkhd-1	trackHeight	848

First atom found is the *"fytp"* as indicated by the specification [19]. As the Container atoms are organized hierarchically *(ie./moov/)*, they in turn have child Container atoms *(ie./moov/trak)* and labels (ie*./moov/mvhd/tkhd/flags)* and this tags also contain values (ie*./moov/mvhd/tkhd/version*, *value:0)*.

TABLE 3. Feature extracted from container atoms information.

PathField	PathFieldValue	PathOrderField	PathOrderFieldValue
ftyp/majorBrand	ftyp/majorBrand = mp42	ftyp-1/mp42	ftyp-1/majorBrand = mp42
ftyp/minorVersion	ftyp/minorVersion = 1	ftyp-1/1	ftyp-1/minorVersion = 1
ftyp/compatibleBrands	ftyp/compatibleBrands = mp41mp42isom	ftyp-1/mp41mp42isom	ftyp-1/compatibleBrands = mp41mp42isom
beam/byteInitial	beam/byteInitial = 28	beam-2/28	beam-2/byteInitial = 28
beam/size	beam/size = 42	beam-2/42	beam-2/size = 42
moov/trak/tkhd/version	moov/trak/tkhd/version = 0	moov-4/trak-2/tkhd-1/0	moov-4/trak-2/tkhd-1/version = 0
moov/trak/tkhd/flags	moov/trak/tkhd/flags = 1	moov-4/trak-2/tkhd-1/1	moov-4/trak-2/tkhd-1/flags = 1
moov/trak/tkhd/trackId	moov/trak/tkhd/trackId = 1	moov-4/trak-2/tkhd-1/1	moov-4/trak-2/tkhd-1/trackId = 1
moov/trak/tkhd/trackWidth	moov/trak/tkhd/trackWidth = 48	moov-4/trak-2/tkhd-1/48	moov-4/trak-2/tkhd-1/trackWidth = 48
moov/trak/tkhd/trackHeight	moov/trak/tkhd/trackHeight = 848	moov-4/trak-2/tkhd-1/848	moov-4/trak-2/tkhd-1/trackHeight = 848

In this proposal, in order to make the clustering of videos it has been taken into account that the videos are sets of elements that contain the following features: ''PathField'', ''PathFieldValue'', ''PathOrderField''and ''PathOrderFieldValue''. ''PathField'' is defined as the union of the Path and Field tags separated by the character ('/'), ''PathFieldValue'' is defined as the union of the Path and Field tags separated by the '/' character and the Value tag separated by the '=' character is added, ''PathOrderField'' is defined as the union of the PathOrder tags and the Field tag separated by the '/' character and ''PathOrderFieldValue'' is the union of the PathOrder tags and the Field tag separator by the '/' character, then the Value tag separated by '=' is added. Table 3 shows an example of the values of these features for each row in Table 2.

The representation proposed in [20] only uses the PathOrderFieldValue universe, however all possible combinations of universes must be taken into account since good results are obtained as will be detailed later in the experimentation chapter of this article.

B. CLUSTERING TECHNIQUES

Clustering is an initial and fundamental step in data analysis. It is an unsupervised classification of patterns into groups or clusters. Intuitively, patterns within a valid cluster are more similar to each other and dissimilar when compared to a pattern belonging to other cluster. Clustering is useful in several fields such as statistics, machine learning, pattern analysis and many other fields. Clustering can be classified into five major types: Partitioned, Hierarchical, Density-Based, Grid-Based and Model-Based methods. In this work, two clustering algorithms have been used to classify videos, which are: a density algorithm called OPTICS and an hierarchical algorithm. Both are detailed in the subsequent sections.

1) DENSITY-BASED METHOD

This was introduced to discover clusters of arbitrary shape. It is based on the fact that within each cluster there is a typical density of points and this density is higher than outside the cluster. Outside points with lower density are recognized as noise points. One of the most commonly known algorithm in this category is, OPTICS: Ordering Points To Identify the Clustering Structure [23]. It was presented by Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel and J. Sander.

OPTICS computes an ordering of the points augmented by additional information, i.e. the reachability distance, representing the intrinsic hierarchical cluster structure. The result of OPTICS i.e. the cluster ordering, is displayed by the so-called reachability plots, which are 2D-plots generated as follows: the clustered objects are ordered along the x-axis according to the cluster ordering computed by OPTICS and the reachabilities assigned to each object are plotted along the abscissa [23]. The key idea of density-based clustering is that for each object of a cluster the neighborhood of a given radius (ϵ) has to contain at least a minimum number of objects *(MinPts)*, i.e. the cardinality of the neighborhood has to exceed a threshold. The formal definitions for this notion of a clustering are introduced in [23] as:

- *Definition 1 (Directly Density-Reachable):* Object p is directly density-reachable from object q with respect to ϵ and *MinPts* in a set of objects D if
  1) $p \in N_\epsilon(q)$ (being $N_\epsilon(q)$ the subset of D contained in the ϵ-neighborhood of q).
  2) $Card(N_\epsilon(q)) \geq MinPts$ (where $Card(N)$ denotes the cardinality of the set N).

  The condition $Card(N_\epsilon(q)) \geq MinPts$ is called the *core object condition*. If this condition holds for an object p, then we call p a *core object*. Only from core objects, other objects can be directly density-reachable.
- *Definition 2 (Density-Reachable):* An object p is *density-reachable* from an object q with respect to ϵ and *MinPts* in the set of objects D if there is a chain of objects $p_l, \ldots, p_m$, $p_l = q$, $p_n = p$ such that $p_i \in D$ and p_{i+1} is directly density-reachable from p_i with respect to ϵ and *MinPts*. Density-reachability is the transitive hull of direct density This relation is not symmetric in general. Only core objects can be mutually density-reachable.
- *Definition 3 (Density-Connected):* Object p is density-connected to object q with respect to ϵ and *MinPts* in the set of objects D if there is an object o in D such that both p and q are density-reachable from o with respect to ϵ and *MinPts* in D. Density-connectivity is a symmetric relation. A density-based cluster is now defined as a set

of density-connected objects which is maximal with respect to density-reachability and the noise is the set of objects not contained in any cluster [23].

- *Definition 4 (Core-Distance of an Object p):* Let p be an object from a database D, let ϵ be a distance value, let N_ϵ, (p) be the ϵ-neighborhood of p, let *MinPts* be a natural number and let *MinPts-distance*(p) be the distance from p to its *MinPts* neighbor. Then, the core-distance of p is defined as core-distance$_{\epsilon,MinPts}(p) =$

$$\begin{cases} Undefined, & \text{if } Card_\epsilon(p) < MinPts \\ MinPts - distance(p), & \text{otherwise} \end{cases} \quad (1)$$

The core-distance of an object p is simply the smallest distance ϵ between p and an object in its ϵ-neighborhood such that p would be a core object with respect to ϵ if this neighbor is contained in $N_\epsilon(p)$. Otherwise, the core-distance is undefined.

2) HIERARCHICAL METHOD

Hierarchical clustering techniques proceed by either a series of successive mergers or a series of successive divisions. In this way these methods can be classified into two principal groups:

- Agglomerative hierarchical methods (bottom-up approach): They start from the individual elements and add them in groups.
- Divisive hierarchical methods (top-down approach): They start from the set of elements and divide it successively until to reach the individual elements.

The agglomerative algorithms that are used always have the same structure and only differ in the way the distances between groups are calculated. Their structure is shown in Algorithm 1.

Algorithm 1 Agglomerative Algorithms

1. Start with N clusters, each containing a single entity and an $N \times N$ symmetric matrix of distances (or similarities) $\boldsymbol{D} = (d_{ik})$;
2. Select the two closest elements in the distance matrix and form with them a class;
3. Replace the two elements used in (2) to define the class by a new element that represents the built class. The distances between this new element and the are calculated using one of the criteria discussed below;
4. Go back to (2) and repeat (2) and (3) until we have all the elements grouped into a unique class;

There are different criteria to calculate the distances between groups.The most common types are for example single linkage, complete linkage, weighted average, etc. [24]. In this work an agglomerative hierarchical clustering algorithm is used and the selected criteria is the weighted average. The results of both agglomerative and divisive methods can be displayed in a two-dimensional diagram known as a dendrogram that shows the mergers or divisions that have been made at successive levels. Once the dendrogram has been obtained, clusters must be extracted.

IV. EXPERIMENTS AND RESULTS

A. DATASETS

In order to carry out the experiments, two datasets have been used, specifically are: VISION dataset [25] and ACID dataset [4], because they are the most complete and current datasets available for forensic analysis in multimedia videos and they are the most recent in the literature. With both datasets we cover the largest number of digital videos available in our society, that is, videos from mobile devices, digital cameras and videos from the main messaging platforms (Whatsaap) and video sharing (Youtube). The VISION dataset is currently composed by 34427 images and 1914 videos, both in the native format and in their social version (Facebook, YouTube, and WhatsApp are considered), from 35 portable devices of 11 major brands. The video-ACID database contains over 12000 videos from 46 physical devices representing 36 unique camera models. Videos in this database are hand collected in a diversity of real-world scenarios are unedited and have known and trusted provenance. For this work, a subset of these videos have been used, those belonging to the eval subset. In our experimentation we have select two samples, one from VISION dataset (sample 1) and other from ACID dataset (sample 2). Tables 4 and 5 show a description of VISION and ACID dataset samples.

B. EXPERIMENTAL CONDITIONS

The following considerations have been taken into account to carry out the experiments: first of all, it is necessary to keep in mind that Field's tag is not always valid to identify the source because they have specific values that depend on the video itself, in the case of tags related to the creation date, duration, etc. The following Container atoms have been removed: *modificationTime*, *creationTime*, *entryCount*, *sampleCount*, *freeSpace*, *duration*. Secondly, as universe all possible representations of the labels of Container atoms have been defined, specifically: PathField, PathFieldValue, PathOrderField and PathOrderFieldValue. A summary of the experimental conditions is shown below in Table 6.

C. RESULTS

In order to choose the best representation of the dataset in different clusters, and the best metric, the Silhouette Coefficient has been used. The representation and measure with the highest Silhouette Coefficient will be the most likely to be correctly separated. The Silhouette Coefficient has been widely used in other multimedia forensic analysis works [26]–[28]. The Silhouette Coefficient measures the average distance from one point to all other points in the same group (cohesion a_j), on the other hand it also measures the

TABLE 4. Composition of the sample of VISION.

Brand	Model	# Videos
Apple	iPad2	16
	Ipad mini	16
	iPhone 4	19
	iPhone 4S	28
Asus	Zenphone 2 Laser	19
Huawei	Ascend G6-U10	19
	Honor 5C NEM-L51	19
	P8 GRA-L09	19
	P9 EVA-L09	19
	P9 Lite VNS-L31	19
Lenovo	Lenovo P70-A	19
LG	D290	19
Microsoft	Lumia 640 LTE	10
OnePlus	A3000	19
	A3003	19
Samsung	Galaxy S III Mini GT-I8190	16
	Galaxy S III Mini GT-I8190N	22
	Galaxy S3 GT-I9300	19
	Galaxy S4 mini GT-I9195	19
	Galaxy S5 SM-G900F	19
	Galaxy Tab 3 GT-P5210	37
	Galaxy Tab A SM-T555	16
	Galaxy Trend Plus GT-S7580	16
Sony	Xperia Z1 Compact D5503	19
Wiko	Ridge 4G	11
WhatsApp	WhatsApp	644
Xiaomi	Redmi Note 3	19
Youtube	Youtube	622

TABLE 5. Composition of the sample of ACID.

Brand	Model	Device	# Videos
Apple	iPhone 8 plus	M00	223
Asus	Zenfoe 4 Laser	M01	239
Canon	VIXIA HF R800	M06	25
Google	Pixel 1	M10	25
	Pixel 2	M11	25
Huawei	Honor 6X Pixel 2	M12	25
	Honor Mate SE 2	M13	25
Kodak	Ektra	M15	25
LG	Q6	M16	25
	X Charge	M17	25
Moto	E4	M18	25
	G5 plus	M19	25
Nikon	Coolpix S33	M20	25
	Coolpix S3700	M21	25
	Coolpix S7000	M22	25
Olympus	Stylus Tough TG-860	M24	25
Samsung	Galaxy J7 Pro	M27	25
	Galaxy S5	M29	25
	Galaxy S7	M30	25
	Galaxy Tab A	M31	25

average distance of a point from one of the groups to all other nearby groups (b_j separation).

Definition 5: The Silhouette Coefficient is a measure of the consistency of the clusters. It measures both the cohesion and separation of the clusters. Let C_i, $i = 1, \ldots, k$ be the set of clusters (or a cluster configuration). For $i \in C_i$, let

$$a(i) = \frac{1}{|C_i| - 1} \sum_{j \in C_i, j \neq i} d(i, j) \tag{2}$$

be the mean distance between i and all other data points in the same cluster, where $d(i, j)$ is the distance between i and j in the cluster C_i,

$$b(i) = \min_{k \neq i} \frac{1}{|C_k|} \sum_{j \in C_k} d(i, j) \tag{3}$$

to be the smallest mean distance of i to all points in any other cluster, of which i is not a member. The silhouette coefficient is defined as:

$$s(i) = \begin{cases} 1 - a(i)/b(i), & \text{if } a(i) < b(i) \\ 0, & \text{if } a(i) = b(i) \\ a(i)/b(i) - 1, & \text{if } a(i) > b(i) \end{cases} \tag{4}$$

In Table 7 the maximum (for any metric) Silhouette Coefficient in each the two samples of study is shown. In addition, Tables 8 and 9 show the result of the 4 best metrics that have

TABLE 6. Configuration of the experiment.

Metric	Universe	Class	Algorithm	Filtering
Braycurtis	PathField	Brand	Hierarchical	Yes
Canberra	PathOrderField	Model	OPTICS	
Cityblock	PathFieldValue	Device	Silhouette Coefficient	
Correlation	PathOrderFieldValue			
Cosine				
Dice				
Euclidean				
Hamming				
Jaccard				
l1				
l2				
Manhattan				
Matching				
Minkowski				
Rogerstanimoto				
Sokalmichener				
Sokalsneath				

TABLE 7. Maximum average Silhouette coefficient for any metric for samples 1 and 2 respectively.

Universe	Class	Silhouette Coefficient	
		VISION dataset	ACID dataset
PathField	Brand	0.042925	0.532204
PathField	Device	-0.044301	0.405100
PathField	Model	-0.104576	0.405133
PathFieldValue	Brand	0.490063	0.586225
PathFieldValue	Device	0.492146	0.653571
PathFieldValue	Model	0.465963	0.653581
PathOrderField	Brand	0.747372	0.907504
PathOrderField	Device	0.666355	0.805020
PathOrderField	Model	0.609535	0.805028
PathOrderFieldValue	Brand	0.585747	0.814979
PathOrderFieldValue	Device	0.538426	0.795484
PathOrderFieldValue	Model	0.485761	0.795490

given good results in both datasets that have been calculated using the Silhouette Coefficient.

D. EVALUATION FOR CLUSTERING PERFORMANCE

Clustering comparison measures play an important role in cluster analysis. Numerous measures for comparing clusters have been proposed [29]. To measure the performance of the clustering, several metrics have been used to compare the predicted groups with the actual classes of the videos. In particular, the Homogeneity, Completeness and Rand Index (RI) scores metrics have been used. A clustering result satisfies homogeneity if all of its clusters contain only data points which are members of a single class. A clustering result satisfies completeness if all the data points that are members of a given class are elements of the same cluster. Both scores have positive values between 0.0 and 1.0.

The Rand Index (RI) computes a similarity measure between two clusters by considering all pairs of samples and counting pairs that are assigned in the same or different clusters in the predicted and true clusters. The Rand Index is related with accuracy. The accuracy of the predicted partition to correctly bound two points in the same cluster, or not, depending on the real partition. This RI is normalized, with values on range [0, 1], however, does not meet the ''constant baseline'' property.

E. RESULTS OF HIERARCHICAL CLUSTER ALGORITHM

All executions have been complete with the different configurations, shown in Tables 8 and 9, for each of the datasets used in this work. Table 10 shows the summary of the experimental conditions that have been used in the Hierarchical cluster algorithm.

1) RESULTS FOR THE SAMPLE OF VISION DATASET

The maximum RI value for the original data set is with PathField universe, Euclidean metric and Threshold = 1.132 value, producing the clusters are shown in Figure 1 and Table 11 where it can be seen that the number of clusters that have been identified, are 17 of the 13 models available in the VISION dataset. Videos from the YouTube and WhatsApp platforms have been identified almost entirely in a cluster each. The Apple brand identifies several clusters having a different behavior than the other brands. Other devices from different brands, on the contrary, show no difference with this representation, as the Asus Zenfone, producing videos exactly like the ones from Huawei's Honor 5c and P8. It is always impossible to distinguish an Asus video from a Huawei one with this representation of the data, the OnePlus brand is also distinguishable, as the LG or the Wiko brands.

Detail of each of the models that make up the dataset can be seen in Figure 2 and Table 12. They show the result in the case of the models of each of the devices that belong to the dataset.

Finally, the RI value of this configuration is 0.8839, likewise, it can be seen that this configuration has a homogeneity of 0.9195 and an integrity of 0.7970 for VISION's dataset.

TABLE 8. Top metrics for the sample of VISION.

Universe	Class	Metric	Result
PathField	Brand	Euclidean	0.0429
PathField	Brand	Correlation	-0.0029
PathField	Brand	Rogers-Tanimoto	0.0047
PathField	Brand	Sokal-Sneath	0.0103
PathField	Device	Euclidean	-0.0443
PathField	Device	Correlation	-0.0504
PathField	Device	Rogers-Tanimoto	-0.0503
PathField	Device	Sokal-Sneath	-0.0488
PathField	Model	Euclidean	-0.1045
PathField	Model	Correlation	-0.1123
PathField	Model	Rogers-Tanimoto	-0.1127
PathField	Model	Sokal-Sneath	-0.1112
PathFieldValue	Brand	Euclidean	0.3073
PathFieldValue	Brand	Correlation	0.4900
PathFieldValue	Brand	Rogers-Tanimoto	0.4782
PathFieldValue	Brand	Sokal-Sneath	0.4021
PathFieldValue	Device	Euclidean	0.3042
PathFieldValue	Device	Correlation	0.4921
PathFieldValue	Device	Rogers-Tanimoto	0.4860
PathFieldValue	Device	Sokal-Sneath	0.4137
PathFieldValue	Model	Euclidean	0.2919
PathFieldValue	Model	Correlation	0.4659
PathFieldValue	Model	Rogers-Tanimoto	0.4586
PathFieldValue	Model	Sokal-Sneath	0.3940
PathOrderField	Brand	Euclidean	0.7473
PathOrderField	Brand	Correlation	0.7253
PathOrderField	Brand	Rogers-Tanimoto	0.7211
PathOrderField	Brand	Sokal-Sneath	0.7329
PathOrderField	Device	Euclidean	0.6663
PathOrderField	Device	Correlation	0.6267
PathOrderField	Device	Rogers-Tanimoto	0.6337
PathOrderField	Device	Sokal-Sneath	0.6480
PathOrderField	Model	Euclidean	0.6095
PathOrderField	Model	Correlation	0.5647
PathOrderField	Model	Rogers-Tanimoto	0.5705
PathOrderField	Model	Sokal-Sneath	0.5849
PathOrderFieldValue	Brand	Euclidean	0.3838
PathOrderFieldValue	Brand	Correlation	0.5857
PathOrderFieldValue	Brand	Rogers-Tanimoto	0.5711
PathOrderFieldValue	Brand	Sokal-Sneath	0.4820
PathOrderFieldValue	Device	Euclidean	0.3545
PathOrderFieldValue	Device	Correlation	0.5379
PathOrderFieldValue	Device	Rogers-Tanimoto	0.5384
PathOrderFieldValue	Device	Sokal-Sneath	0.4673
PathOrderFieldValue	Model	Euclidean	0.3230
PathOrderFieldValue	Model	Correlation	0.4857
PathOrderFieldValue	Model	Rogers-Tanimoto	0.4823
PathOrderFieldValue	Model	Sokal-Sneath	0.4239

Summary of these results for brand is shown in Table 13. Table 14 shows detail of the result grouped by model.

2) RESULTS FOR THE SAMPLE OF ACID DATASET

As it can be seen in Figure 3 and Table 15 there are 11 clusters of the 11 brands that belong to the dataset. In this case, unlike the VISION dataset, the Apple brand is correctly classified into a single cluster. The LG and Moto brands cannot be distinguished with this representation. As for digital cameras, it can be seen that the Canon and Olimpus brand are correctly classified, however the same is not true with the Kodak brand that cannot be distinguished from the Samsung brand. The performance execution is shown in Table 16.

Figure 4 and Table 17 show the detail of the clusters generated by the models that belong to the sample of ACID dataset. Table 18 shows the result of the algorithm execution.

TABLE 9. Top metrics of the sample of ACID.

Universe	Class	Metric	Result
PathField	Brand	Euclidean	0.5322
PathField	Brand	Correlation	0.5238
PathField	Brand	Rogers-Tanimoto	0.5238
PathField	Brand	Sokal-Sneath	0.5243
PathField	Device	Euclidean	0.4046
PathField	Device	Correlation	0.4048
PathField	Device	Rogers-Tanimoto	0.4050
PathField	Device	Sokal-Sneath	0.4050
PathField	Model	Euclidean	0.4051
PathField	Model	Correlation	0.4048
PathField	Model	Rogers-Tanimoto	0.4050
PathField	Model	Sokal-Sneath	0.4050
PathFieldValue	Brand	Euclidean	0.4127
PathFieldValue	Brand	Correlation	0.5862
PathFieldValue	Brand	Rogers-Tanimoto	0.5781
PathFieldValue	Brand	Sokal-Sneath	0.5349
PathFieldValue	Device	Euclidean	0.4521
PathFieldValue	Device	Correlation	0.6535
PathFieldValue	Device	Rogers-Tanimoto	0.6448
PathFieldValue	Device	Sokal-Sneath	0.5999
PathFieldValue	Model	Euclidean	0.4521
PathFieldValue	Model	Correlation	0.6535
PathFieldValue	Model	Rogers-Tanimoto	0.6448
PathFieldValue	Model	Sokal-Sneath	0.5999
PathOrderField	Brand	Euclidean	0.8945
PathOrderField	Brand	Correlation	0.8921
PathOrderField	Brand	Rogers-Tanimoto	0.8934
PathOrderField	Brand	Sokal-Sneath	0.9075
PathOrderField	Device	Euclidean	0.8045
PathOrderField	Device	Correlation	0.8050
PathOrderField	Device	Rogers-Tanimoto	0.8049
PathOrderField	Device	Sokal-Sneath	0.8047
PathOrderField	Model	Euclidean	0.8045
PathOrderField	Model	Correlation	0.8050
PathOrderField	Model	Rogers-Tanimoto	0.8049
PathOrderField	Model	Sokal-Sneath	0.8047
PathOrderFieldValue	Brand	Euclidean	0.6659
PathOrderFieldValue	Brand	Correlation	0.8205
PathOrderFieldValue	Brand	Rogers-Tanimoto	0.8149
PathOrderFieldValue	Brand	Sokal-Sneath	0.7622
PathOrderFieldValue	Device	Euclidean	0.6196
PathOrderFieldValue	Device	Correlation	0.7954
PathOrderFieldValue	Device	Rogers-Tanimoto	0.7896
PathOrderFieldValue	Device	Sokal-Sneath	0.7374
PathOrderFieldValue	Model	Euclidean	0.6196
PathOrderFieldValue	Model	Correlation	0.7954
PathOrderFieldValue	Model	Rogers-Tanimoto	0.7896
PathOrderFieldValue	Model	Sokal-Sneath	0.7374

TABLE 10. Configuration of the hierarchical experiment.

Hierarchical's parameters	Universe	Metric	Threshold	Criterion	Linkage
Values	PathField	Euclidean	1.132	Linkage	Single

In the results shown above for this dataset and with the configuration selected, it can be seen that RI values are obtained for both the brand and the model greater than 0.80 specifically, 0.8128 and 0.8233 respectively. The homogeneity in the case of the brand is higher than in the case of the model because fewer clusters are obtained than models. The opposite occurs with the integrity that reaches 1.0 in the case of the model.

Finally, Tables 19 and 20 show the comparative results for both samples using the Hierarchical Clustering algorithm.

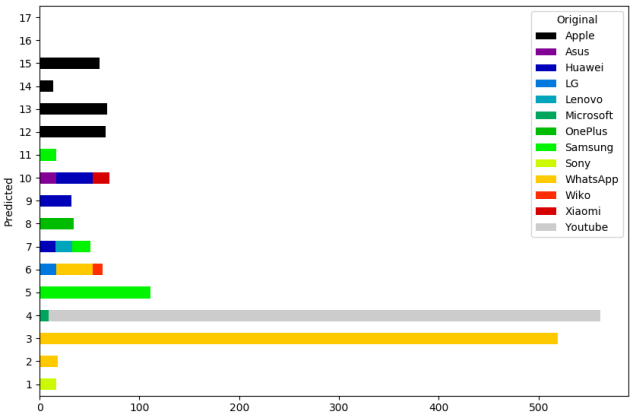

FIGURE 1. Result of hierarchical clustering algorithm grouped by brand using VISION dataset.

TABLE 11. Number of videos per cluster grouped by brand using hierarchical clustering algorithm of VISION dataset.

Cluster	Brand	Count
1	Sony	17
2	WhatsApp	18
3	WhatsApp	519
4	Microsoft	9
	Youtube	553
5	Samsung	111
6	LG	17
	WhatsApp	36
	Wiko	10
7	Huawei	16
	Lenovo	17
	Samsung	18
8	OnePlus	34
9	Huawei	32
10	Asus	17
	Huawei	36
	Xiaomi	17
11	Samsung	17
12	Apple	66
13	Apple	68
14	Apple	14
15	Apple	60
16	Apple	1
17	Youtube	1

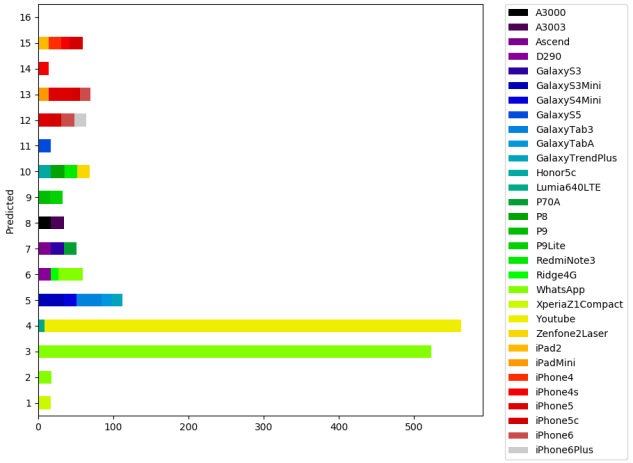

FIGURE 2. Result of the hierarchical clustering algorithm grouped by model using VISION dataset.

F. RESULTS OF OPTICS ALGORITHM

The all configurations shown in Table 8 and Table 9 had been used to run OPTICS. The remaining parameters are *minPoints* and epsilon, ϵ values. The first will be fixed to 5, as OPTICS main difference with Hierarchical Clustering is its ability to

TABLE 12. Number of videos per cluster grouped by model using hierarchical clustering algorithm of VISION dataset.

Cluster	Model	Count
1	XperiaZ1 Compact	17
2	WhatsApp	18
3	WhatsApp	523
4	Lumia640LTE	9
	Youtube	554
5	GalaxyS3Mini	34
	GalaxyS4Mini	17
	GalaxyTab3	33
	GalaxyTabA	14
	GalaxyTrendPlus	14
6	D290	17
	Ridge4G	10
	WhatsApp	32
7	Ascend	17
	GalaxyS3	17
	P70A	17
8	A3000	17
	A3003	17
9	P9	16
	P9Lite	17
10	Honor5c	17
	P8	17
	P9	1
	RedmiNote3	17
	Zenfone2Laser	17
11	GalaxyS5	17
12	iPhone5	14
	iPhone5c	17
	iPhone6	17
	iPhone6Plus	16
13	iPadMini	14
	iPhone5	31
	iPhone5c	11
	iPhone6	14
14	iPhone4s	14
15	iPad2	14
	iPhone4	17
	iPhone4s	11
	iPhone5c	17
16	iPhone6Plus	1

TABLE 13. Results of run the hierarchical clustering algorithm grouped by brand of sample of VISION dataset.

#Brands	#Clusters	RI	Homogeneity	Completeness
13	17	0.8839517587	0.9195359995	0.7970734665

TABLE 14. Results of run the hierarchical clustering algorithm grouped by model of sample of VISION dataset.

#Models	#Clusters	RI	Homogeneity	Completeness
31	16	0.9058522498	0.8040815705	0.9030898189

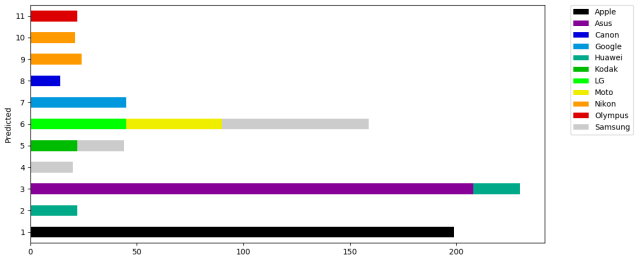

FIGURE 3. Result of executing hierarchical clustering algorithm grouped by brand for the sample of ACID dataset.

TABLE 15. Number of videos per cluster grouped by brand using hierarchical clustering algorithm of ACID dataset.

#Cluster	Brand	Count
1	Apple	199
2	Huawei	22
3	Asus	208
	Huawei	22
4	Samsung	20
5	Kodak	22
	Samsung	22
6	LG	45
	Moto	45
	Samsung	69
7	Google	45
8	Canon	14
9	Nikon	24
10	Nikon	21
11	Olympus	22

ignore the noise. ϵ, however, has been varied. After several executions with different values of the ϵ parameter, it has been concluded that the value $\epsilon = 0.01$ offers good results in both datasets. Table 21 shows the summary of the experimental conditions of the OPTICS algorithm.

TABLE 16. Result of run the hierarchical clustering algorithm grouped by brand for the sample of ACID dataset.

#Brands	#Clusters	RI	Homogeneity	Completeness
11	11	0.8128426754	0.8324380092	0.8923993328

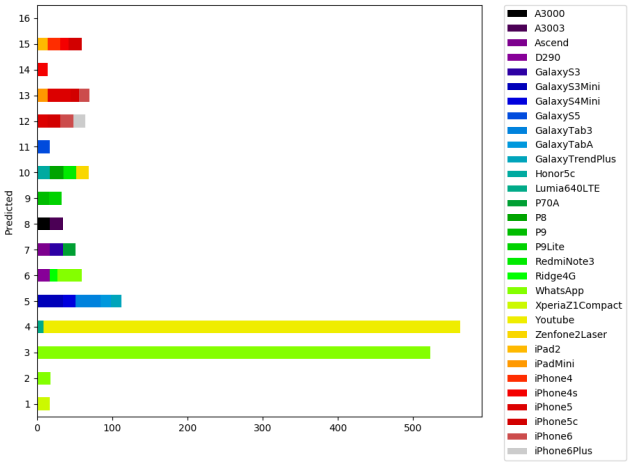

FIGURE 4. Result of executing hierarchical clustering algorithm grouped by model for the sample of ACID dataset.

TABLE 17. Number of videos per cluster grouped by model using hierarchical clustering algorithm of ACID dataset.

#Cluster	Model	Count
1	iPhone8plus	199
2	MateSE	22
3	Honor6X	22
	Zenfone3Laser	209
4	E4	22
	G5Plus	22
	GalaxyJ7Pro	22
	GalaxyS7	22
	GalaxyTabA	22
	Q6	22
	XCharge	22
5	GalaxyS5	23
6	Ektra	23
	J5-6	23
7	Pixel1	23
	Pixel2	22
8	VIXIAH FR800	14
9	CoolpixS33	22
10	Coolpix S7000	22
11	Stylus	22

TABLE 18. Result of executing hierarchical clustering algorithm grouped by Model for the sample of ACID dataset.

#Models	#Clusters	RI	Homogeneity	Completeness
20	11	0.8233019504	0.778805082778	1.0

TABLE 19. Hierarchical clustering algorithm grouped by brand for both samples.

Parameter	VISION	ACID
#Brands	13	11
#Clusters	17	11
RI	0.8839517587	0.8128426754
Homogeneity	0.9195359995	0.8324380092
Completeness	0.7970734665	0.8923993328

1) RESULTS OF OPTICS ALGORITHM FOR THE SAMPLE OF VISION DATASET

As it can be seen in Figure 5 and Table 22 the algorithm has generated 25 clusters of the 13 brands that belong to the sample of VISION's dataset. As with the hierarchical clustering algorithm, the Apple brand needs several

TABLE 20. Hierarchical clustering algorithm grouped by model for both samples.

Parameter	VISION	ACID
#Brands	16	11
#Clusters	31	11
RI	0.793033503	0.8233019504
Homogeneity	0.804081570	0.7788050827
Completeness	0.903089818	1.0

TABLE 21. Configuration of the OPTICS experiment that performed the best result.

OPTICS's parameters	Values
Universe	PathOrderField
Metric	Roger-Stanimoto
Epsilon	0.01
MinPoints	5

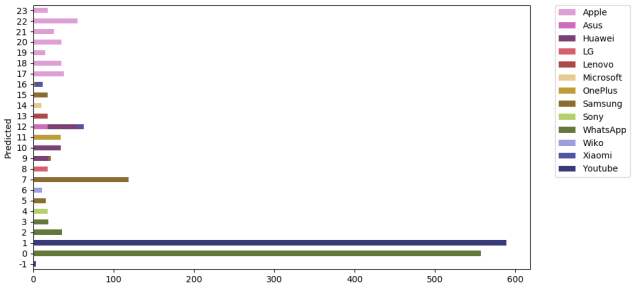

FIGURE 5. Result of run OPTICS algorithm grouped by brand for the sample of VISION dataset.

TABLE 22. Number of videos per cluster grouped by brand using OPTICS algorithm of VISION dataset.

#Cluster	Brand	Count	#Cluster	Brand	Count
-1	Apple	1	12	Asus	18
	Youtube	2		Huawei	37
0	WhatsApp	557		Xiaomi	8
1	Youtube	589	13	Lenovo	18
2	WhatsApp	36	14	Microsoft	10
3	WhatsApp	19	15	Samsung	18
4	Sony	18	16	OnePlus	2
5	Samsung	16		Xiaomi	10
6	Wiko	11	17	Apple	38
7	Samsung	119	18	Apple	35
8	LG	18	19	Apple	15
9	Huawei	19	20	Apple	35
	Samsung	3	21	Apple	26
10	Huawei	34	22	Apple	55
11	OnePlus	34	23	Apple	18

clusters to identify itself, the good news is that in those clusters there is no mix of another brand. Videos from YouTube or WhatsApp are mostly classified in a cluster by model. Therefore, the algorithm is capable of grouping native videos from mobile devices and also videos that have been downloaded from online platforms such as YouTube or WhatsApp. The result of the execution of the algorithm

TABLE 23. Result of run the OPTICS algorithm in the sample of VISION dataset group by Brand.

#Brands	#Clusters	RI	Homogeneity	Completeness
13	25	0.8930095982	0.9759981843	0.7737539175

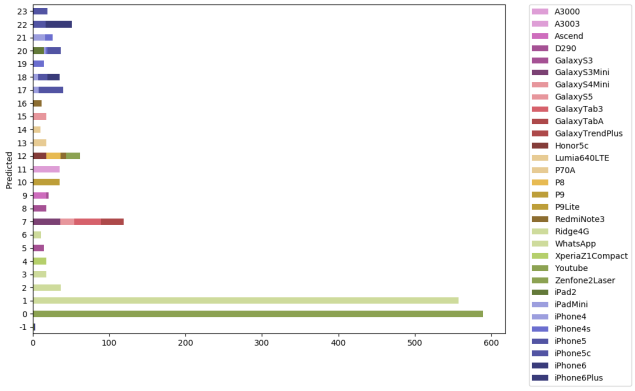

FIGURE 6. Result of the clustering with OPTICS algorithm grouped by model for the sample of VISION dataset.

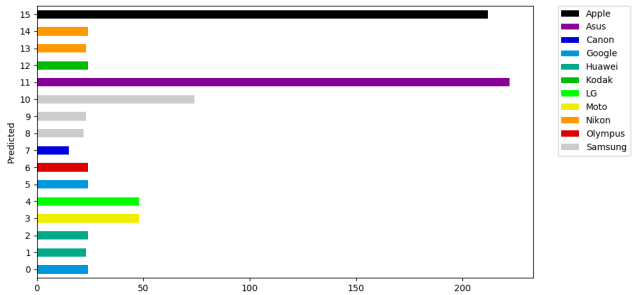

FIGURE 7. Result of the clustering with OPTICS algorithm grouped by brand for the sample of ACID dataset.

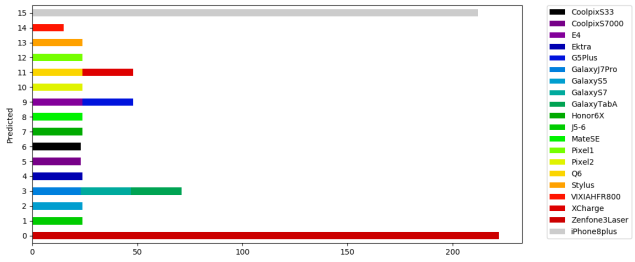

FIGURE 8. Result of executing OPTICS algorithm grouped by model for the sample of ACID dataset.

can be seen in detail in Table 23. With the OPTICS algorithm the parameters RI and Homogeneity and integrity are very similar to the hierarchical algorithm. It can be concluded that the selected algorithm does not interfere with the identification but with the configuration selected in each algorithm.

The detail by model is shown in Figure 6 and Table 24. The result of the execution of the algorithm can be seen in Table 25.

TABLE 24. Number of videos per cluster grouped by model using OPTICS algorithm of VISION dataset.

#Cluster	Model	Count	#Cluster	Model	Count
-1	Youtube	2	14	Honor5c	18
	iPhone6Plus	1		P8	18
0	Youtube	589		P9	1
1	WhatsApp	557		RedmiNote3	7
2	WhatsApp	37		Zenfone2Laser	18
3	WhatsApp	18	15	GalaxyS5	18
4	XperiaZ1 Compact	18	16	A3003	1
				RedmiNote3	11
5	GalaxyS3	15	17	iPadMini	8
6	Ridge4G	11		iPhone5	32
7	GalaxyS3Mini	36	18	iPadMini	7
	GalaxyS4Mini	18		iPhone5c	12
	GalaxyTab3	35		iPhone6	16
	GalaxyTabA	15	19	iPhone4s	15
	GalaxyTrend+	15	20	iPad2	15
8	D290	18		iPhone4	2
9	Ascend	18		iPhone4s	2
	GalaxyS3	3		iPhone5c	18
10	P9	17	21	iPhone4	16
	P9Lite	18		iPhone4s	10
11	A3000	18	22	iPhone5	17
	A3003	17		iPhone6	17
12	P70A	18		iPhone6Plus	17
13	Lumia640LTE	10	23	iPhone5c	19

TABLE 25. Result of run the OPTICS algorithm grouped by model for the sample of VISION dataset.

#Models	#Clusters	RI	Homogeneity	Completeness
31	25	0.9210856392	0.8758900022	0.8950395170

TABLE 26. Number of videos per cluster grouped by brand using OPTICS algorithm of ACID dataset.

#Cluster	Brand	Count	#Cluster	Brand	Count
0	Google	24	8	Samsung	22
1	Huawei	23	9	Samsung	23
2	Huawei	24	10	Samsung	74
3	Moto	48	11	Asus	222
4	LG	48	12	Kodak	24
5	Google	24	13	Nikon	23
6	Olympus	24	14	Nikon	24
7	Canon	15	15	Apple	212

TABLE 27. Result of run the OPTICS algorithm grouped by brand for the sample of ACID dataset.

#Brands	#Clusters	RI	Homogeneity	Completeness
11	16	0.9429572989	1.0	0.8934425333

2) RESULTS OF OPTICS ALGORITHM FOR THE SAMPLE OF ACID DATASET

In Figure 7 and Table 26 they can be seen that the algorithm has originated 16 clusters of the 11 marks available in the sample of ACID dataset. The classification is correct both in videos originated by mobile devices and in videos generated by digital cameras. This algorithm has better results than the Hierarchical algorithm. The detail of the execution result can be seen in Table 27.

TABLE 28. Number of videos per cluster grouped by model using OPTICS algorithm of ACID dataset.

#Cluster	Model	Count	#Cluster	Model	Count
0	Zenfone3Laser	222	8	MateSE	24
1	J5-6	24	9	E4	24
2	GalaxyS5	24	9	G5Plus	24
3	GalaxyJ7Pro	23	10	Pixel2	24
3	GalaxyS7	24	11	Q6	24
3	GalaxyTabA	24	11	XCharge	24
4	Ektra	24	12	Pixel1	24
5	CoolpixS7000	23	13	Stylus	24
6	CoolpixS33	23	14	VIXIAHFR800	15
7	Honor6X	24	15	iPhone8plus	212

TABLE 29. Result of run the OPTICS algorithm grouped by model for the sample of ACID dataset.

#Models	#Clusters	RI	Homogeneity	Completeness
16	20	0.9687571912	0.9313286618	1.0

TABLE 30. Algorithm OPTICS grouped by brand.

Parameter	VISION	ACID
#Brands	13	11
#Clusters	25	16
RI	0.8930095982	0.9429572989
Homogeneity	0.9759981843	1.0
Completeness	0.7737539175	0.8934425333

TABLE 31. Algorithm OPTICS grouped by Model.

Parameter	VISION	ACID
#Models	31	16
#Clusters	25	20
RI	0.9210856392	0.9687571912
Homogeneity	0.8758900022	0.9313286618
Completeness	0.8950395170	1.0

The classification by models can be visualized in Figure 8 and Table 28. In Table 29 the detail of the result of the execution group by model is shown.

Finally we show two tables for comparative purposes of the experiments using OPTICS algorithm. The results grouped by brand for the selected samples (sample of VISION dataset and sample of ACID dataset) can be seen in Table 30. On the other hand the comparative results grouped by model for the two selected samples can be seen in Table 31.

V. CONCLUSION

This work has shown how the information of the video files can be exploited to group videos by data source, without prior training of a classifier. In the literature currently available there is a great shortage in the investigation of the source of video acquisition that uses the structure of the video container to obtain the characteristics. An essential point of the proposed methodology has been the correct acquisition of data for further processing and processing. With a good preliminary acquisition, the subsequent treatment through the use of classification algorithms has been effective in determining through the use of Data Mining techniques the final clustering of the same. The proposed methodology has

been validated through two sets of data to which it has been applied with the same selection of parameters in order to obtain comparable results. The data sets used have been obtained by sampling on the two most current databases in the literature. The databases contain videos from various sources: native videos from mobile devices, native videos from digital cameras and videos that have been downloaded from platforms such as WhatsApp and YouTube. It has been considered to obtain sufficiently significant samples to carry out the study. The proposed methodology is general enough to be able to apply it and adapt it to other types of data, as well as apply other classification techniques present in Multivariate Analysis (non-hierarchical classification techniques, use of methods based on statistical models [24], among others). As has been seen in the numerical results obtained from the samples, the proposed clustering algorithms have provided good results from the perspective of the classification. The usage of simple algorithms was also proven effective separating video files by brand. The results were positive, and an algorithm was proven to be able to correctly group videos in homogeneous clusters per brand, even if too many clusters appeared.

ACKNOWLEDGMENT

This project is promoted under the specialty of Industrial Ph.D. in collaboration with Securitas Direct.

REFERENCES

[1] D. R. Hayes, *A Practical Guide to Computer Forensics Investigations*. London, U.K.: Pearson, 2015.

[2] M. C. Stamm, M. Wu, and K. J. R. Liu, "Information forensics: An overview of the first decade," *IEEE Access*, vol. 1, pp. 167–200, 2013.

[3] S. Milani, K. M. Fontani, P. Bestagini, M. Barni, A. Piva, M. Tagliasacchi, and K. S. Tubaro, "An overview on video forensics," in *Proc. 20th Eur. Signal Process. Conf. (EUSIPCO)*, Bucharest, Romania, vol. 1, Aug. 2012, pp. 1229–1233.

[4] B. Hosler, O. Mayer, B. Bayar, X. Zhao, C. Chen, J. A. Shackleford, and M. C. Stamm, "A video camera model identification system using deep learning and fusion," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Brighton, U.K., May 2019, pp. 8271–8275.

[5] X. Zhao and M. C. Stamm, "Computationally efficient demosaicing filter estimation for forensic camera model identification," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Phoenix, AZ, USA, Sep. 2016, pp. 151–155.

[6] C.-H. Choi, H.-Y. Lee, and H.-K. Lee, "Estimation of color modification in digital images by CFA pattern change," *Forensic Sci. Int.*, vol. 226, nos. 1–3, pp. 94–105, Mar. 2013.

[7] F. Peng and D.-L. Zhou, "Discriminating natural images and computer generated graphics based on the impact of CFA interpolation on the correlation of PRNU," *Digit. Invest.*, vol. 11, no. 2, pp. 111–119, Jun. 2014.

[8] P. Mullan, C. Riess, and F. Freiling, "Forensic source identification using JPEG image headers: The case of smartphones," *Digit. Invest.*, vol. 28, pp. S68–S76, Apr. 2019.

[9] J. Lukas, J. Fridrich, and M. Goljan, "Determining digital image origin using sensor imperfections," *Image Video Commun. Process.*, vol. 5685, Mar. 2005, pp. 249–260.

[10] J. Lukáš, J. Fridrich, and M. Goljan, "Detecting digital image forgeries using sensor pattern noise," *Secur., Steganography, Watermarking Multimedia Contents VIII*, vol. 6072, Feb. 2006, pp. 362–372.

[11] M. Chen, J. Fridrich, and M. Goljan, "Digital imaging sensor identification (further study)," *Proc. Secur., Steganography, Watermarking Multimedia Contents IX*, vol. 6505, Feb. 2007, Art. no. 65050P.

[12] T. Filler, J. Fridrich, and M. Goljan, "Using sensor pattern noise for camera model identification," in *Proc. 15th IEEE Int. Conf. Image Process.*, Oct. 2008, pp. 1296–1299.

[13] Y. Hu, B. Yu, and C. Jian, "Source camera identification using large components of sensor pattern noise," in *Proc. 2nd Int. Conf. Comput. Sci. Appl.*, Jeju-do, South Korea, Dec. 2009, pp. 1–5.

[14] C.-T. Li, "Source camera identification using enhanced sensor pattern noise," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 2, pp. 280–287, Jun. 2010.

[15] A. J. Cooper, "Improved photo response non-uniformity (PRNU) based source camera identification," *Forensic Sci. Int.*, vol. 226, nos. 1–3, pp. 132–141, Mar. 2013.

[16] L. J. G. Villalba, A. L. S. Orozco, R. Ramos López, and J. H. Castro, "Identification of smartphone brand and model via forensic video analysis," *Expert Syst. Appl.*, vol. 55, pp. 59–69, Aug. 2016.

[17] *Coding of Audio-Visual Objects-Part 12: ISO Base Media File Format*, Standard ISO/IEC 14496-12:2015, 2017. [Online]. Available: https://www.iso.org/standard/68960.html

[18] (Sep. 2016). *QuickTime File Format Specification: Overview*. [Online]. Available: https://developer.apple.com/library/archive/documentation/QuickTime/QTFF/QTFFChap1/qtff1.html

[19] T. Gloe, A. Fischer, and M. Kirchner, "Forensic analysis of video file formats," *Digit. Invest.*, vol. 11, pp. S68–S76, May 2014.

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

[21] J. Song, K. Lee, W. Y. Lee, and H. Lee, "Integrity verification of the ordered data structures in manipulated video content," *Digit. Invest.*, vol. 18, pp. 1–7, Sep. 2016.

[22] Group of Analysis, Security and Systems, "Metadata video extraction tool," Dept. Comput. Sci. Eng., Univ. Complutense de Madrid, Madrid, Spain, Tech. Rep. GA013, Aug. 2017.

[23] M. Ankerst, M. Breunig, H.-P. Kriegel, and J. Sander, "OPTICS: Ordering points to identify the clustering structure," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, Philadelphia, PA, USA, Jun. 1999, pp. 49–60.

[24] R. A. Johnson and D. W. Wichern, *Applied Multivariate Statistical Analysis*, 6th ed. Upper Saddle River, NJ, USA: Pearson, 2007.

[25] D. Shullani, M. Fontani, M. Iuliani, O. A. Shaya, and A. Piva, "VISION: A video and image dataset for source identification," *EURASIP J. Inf. Secur.*, vol. 1, no. 15, pp. 1–16, Oct. 2017.

[26] S. Khan and T. Bianchi, "Fast image clustering based on camera fingerprint ordering," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Shanghai, China, Jul. 2019, pp. 766–771.

[27] L. J. García Villalba, A. L. Sandoval Orozco, and J. R. Corripio, "Smartphone image clustering," *Expert Syst. Appl.*, vol. 42, no. 4, pp. 1927–1940, Mar. 2015.

[28] F. Marra, G. Poggi, C. Sansone, and L. Verdoliva, "Blind PRNU-based image clustering for source identification," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 9, pp. 2197–2211, Sep. 2017.

[29] N. X. Vinh, J. Epps, and J. Bailey, "Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance," *J. Mach. Learn. Res.*, vol. 11, pp. 2837–2854, Jan. 2010.

RAQUEL RAMOS LÓPEZ received the Computer Science Engineering degree and the M.S. degree in computer science from the Universidad Complutense of Madrid, where she is currently pursuing the Ph.D. degree in computer engineering with the Group of Analysis, Security and Systems (GASS Group). She is currently working as an Analyst in R& D& I projects in the area of digital video processing with Securitas Direct España, S.A.U. Her main research interest is information security and its applications.

ELENA ALMARAZ LUENGO received the Mathematics degree and the Statistical Sciences and Techniques degree from the University Complutense of Madrid, in 2005 and 2007, respectively, the master's degree in advanced mathematics with specialization in statistics and operations research from the National Distance Education University, in 2010, and the Business and Administration degree from the National Distance Education University, in 2015. She is currently pursuing the Ph.D. degree in mathematics with the University Complutense of Madrid. She is also an Assistant Professor with the Department of Statistic and Operational Research, Faculty of Mathematics Sciences, University Complutense of Madrid. Her main interests are statistic techniques, probability, and information security and applications.

ANA LUCILA SANDOVAL OROZCO received the Computer Science Engineering degree from the Universidad Autónoma del Caribe, Colombia, in 2001, the specialization course in computer networks from the Universidad del Norte, Colombia, in 2006, and the M.Sc. degree in research in computer science and the Ph.D. degree in computer science from the Universidad Complutense de Madrid, Spain, in 2009 and 2014, respectively. She is currently a Postdoctoral Researcher with the Universidad Complutense de Madrid. Her main research interests are coding theory, and information security and its applications.

LUIS JAVIER GARCÍA VILLALBA received the Telecommunication Engineering degree from the Universidad de Málaga, Spain, in 1993, and the Ph.D. degree in computer science from the Universidad Politécnica de Madrid, Spain, in 1999. He was a Visiting Scholar with COSIC (Computer Security and Industrial Cryptography), Department of Electrical Engineering, Faculty of Engineering, Katholieke Universiteit Leuven, Belgium, in 2000. He was a Visiting Scientist with the IBM Research Division (IBM) Almaden Research Center, San Jose, CA, USA, in 2001 and 2002. He is currently an Associate Professor with the Department of Software Engineering and Artificial Intelligence, Universidad Complutense de Madrid (UCM), and the Head of the Complutense Research Group GASS (Group of Analysis, Security and Systems) which is located in the Faculty of Computer Science and Engineering, UCM Campus. His professional experience includes the management of both national and international research projects and both public (Spanish Ministry of R&D, Spanish Ministry of Defence, Horizon 2020 - European Commission, and so on) and private financing (Hitachi, IBM, Nokia, Safelayer Secure Communications, TB Solutions Security, and so on). He is the author or coauthor of numerous international publications is the editor or guest editor of numerous journals, such as *Entropy* MPDI, *Future Generation Computer Systems*, *Future Internet* MDPI, the IEEE Latin America Transactions, *IET Communications*, *IET Networks*, *IET Wireless Sensor Systems*, the *International Journal of Ad Hoc and Ubiquitous Computing*, the *International Journal of Multimedia and Ubiquitous Engineering* (IJMUE), *The Journal of Supercomputing*, and *Sensors* MDPI.