AFRL-RI-RS-TR-2020-199

FORENSIC MODULES FOR THE AUTOMATED ANALYSIS OF PHYSICAL INTEGRITY IN IMAGES (FENCE)

UNIVERSITA' DEGLI STUDI DI FIRENZE (UNIVERSITY OF FLORENCE)

NOVEMBER 2020

FINAL TECHNICAL REPORT

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE

■ **AIR FORCE MATERIEL COMMAND** ■ **UNITED STATES AIR FORCE** ■ **ROME, NY 13441**

NOTICE AND SIGNATURE PAGE

AFRL-RI-RS-TR-2020-199 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE CHIEF ENGINEER:

/ S /
TODD B. HOWLETT
Work Unit Manager

/ S /
JAMES S. PERRETTA
Deputy Chief, Information
Exploitation & Operations Division
Information Directorate

REPORT DOCUMENTATION PAGE	*Form Approved* **OMB No. 0704-0188**

1. REPORT DATE *(DD-MM-YYYY)* NOVEMBER 2020	**2. REPORT TYPE** FINAL TECHNICAL REPORT	**3. DATES COVERED** *(From - To)* MAY 2016 – MAY 2020

4. TITLE AND SUBTITLE FORENSIC MODULES FOR THE AUTOMATED ANALYSIS OF PHYSICAL INTEGRITY IN IMAGES (FENCE)	**5a. CONTRACT NUMBER** FA8750-16-2-0188
	5b. GRANT NUMBER N/A
	5c. PROGRAM ELEMENT NUMBER 62702E
6. AUTHOR(S) D. Baracchi D. Shullani C. Colombo A. Piva. M. Fanfani M. Iuliani	**5d. PROJECT NUMBER** MEDI
	5e. TASK NUMBER 40
	5f. WORK UNIT NUMBER 03

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Universita' Degli Studi Di Firenze (University of Florence) Piazza Di San Marco 4 Firenze 50121, Italy	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Force Research Laboratory/RIGC 525 Brooks Road Rome NY 13441-4505	**10. SPONSOR/MONITOR'S ACRONYM(S)** AFRL/RI
	11. SPONSOR/MONITOR'S REPORT NUMBER AFRL-RI-RS-TR-2020-199

12. DISTRIBUTION AVAILABILITY STATEMENT
Approved for Public Release; Distribution Unlimited. This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
As part of the Defense Advanced Research Project Agency (DARPA) Media Forensics (MediFor) Program this contract was to contribute to the research, development and testing of advanced media forensic techniques to automatically detect and locate media manipulations. A number of techniques were pursued. Three scene based algorithms were researched first. Two techniques for the automatic detection of asymmetric cropping of digital images, one was based on metadata analysis and the second based on exploiting geometrical characteristic of the scene by estimating image projective invariants. The third technique was to detect face splicing in images based on the physical analysis of the imaged scene. While the performance of these scene based methods, under controlled scenarios, has been higher than the state of the art, results in the MediFor Evaluations were not satisfying. The team moved on to developing two novel techniques for unsupervised forensic analysis of video file containers. These two tools achieved the highest Area Under the Curve (AUC) scores in the video manipulations task of MediFor evaluations, while at the same time being among the least computationally expensive algorithms. Lastly, an analysis of Photo Response Non-uniformity (PRNU) on images and videos acquired by smartphones was investigated and in particular countermeasures developed for the challenges smartphones present. These last PRNU techniques were not mature enough to participate in the MediFor evaluations, but are presented in this report. This overall effort contributed positively to the greater MediFor research project and the results are summarized in this report.

15. SUBJECT TERMS
Media forensics, cropping detection, face splicing, video container analysis, PRNU

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** **TODD B. HOWLETT**
a. REPORT U	**b. ABSTRACT** U	**c. THIS PAGE** U	UU	50	**19b. TELEPHONE NUMBER (*Include area code*)** **N/A**

Table of Contents

List of Figures

List of Tables

1 Summary

The FENCE was a project managed by AFRL under the Defense Advanced Research Project Agency (DARPA) Media Forensics (MediFor) Program to design a set of reliable forensic tools based on the visual analysis of image physical properties. The final aim was to design and deploy a set of tools able to detect and locate several kinds of image manipulations working in a completely automatic way. The proposed idea was to design a process that should involve the automatic detection/localization of scene level characteristics that typically require the human assistance.

According to this proposed project path, a set of tools for the automatic detection of asymmetric cropping of digital images were designed first. In particular, we developed two different approaches for cropping detection: (i) a detector based on image meta-data analysis, and (ii) a method exploiting geometrical characteristic of the scene in order to estimate image projective invariants that can be used to assess the integrity of the image. After that, we developed a novel technique to detect face splicing based on the physical analysis of the imaged scene. We started from the hypothesis that a possible splice in the image is noticed when, in the same image, light coefficients computed from different parts of the scene or objects exhibit relevant differences. In particular, we designed tools that, if the image under test contains at least two faces, tries to determine if one of the faces has been spliced by means of the comparison of a set of light-related features. A dissimilarity in these features will expose the presence of a splicing. While the performance of these methods under controlled scenarios, described in the papers (Fanfani, et al., 2020) and (Fanfani, et al., 2019) has been higher than the state of the art, results in the MediFor Evaluations have not been satisfying and these research paths have been discontinued.

Our research activity then shifted towards a second path for the manipulation of digital videos. In particular, we designed two novel techniques for unsupervised forensic analysis of video file containers. The core idea is the fact that different manufacturers, models and software processing produce videos with subtle differences in the file container structure and content. The first algorithm is able to analyze video containers by providing a formal metric to automatically quantify the dissimilarity between two containers. The measure accounts for both the container structure and content, and has proven to be effective in distinguishing videos whose integrity is preserved from videos whose integrity is compromised. After that, we introduced a new container-based method capable not only to provide an indication of integrity, but also to identify the software used to perform a video manipulation. This was achieved by using a decision-tree-based classifier applied to a vectorial representation of the video container structure. Extensive experiments were carried out on publicly available datasets and during MediFor evaluations, showing excellent results for the integrity verification task. The proposed techniques are shown to be able to also automatically detect manipulations that are performed without video re-encoding, which is an unprecedented achievement for a video forensic algorithm. Moreover, the proposed approaches require an extremely small computational cost as opposed to existing techniques based on the video stream analysis. Our two tools achieved the highest AUC scores in the video manipulations task of MediFor evaluations, while at the same time being among the least computationally expensive algorithms.

The third research path involved the analysis of PRNU) on images and videos acquired by smartphones. It is known that PRNU is a unique trace left into the content by the sensor that allows to link an image or video to the originating device. However, smartphones exhibit several peculiarities that does not allow the application of the usual PRNU extraction and detection, thus requiring proper

countermeasures. In particular, we worked on three sub-topics: (i) a method for hybrid camera identification, that uses PRNU from images and videos; (ii) a calibration technique that can be reliably used to estimate the scale factors relating different acquisition modalities (image, video, and stabilized video) of a given device and a laboratory setup to deeply understand Electronic Image Stabilization (EIS). Then, (iii) a method exploiting deep neural networks to register PRNU signal under small scale and rotation transformations is presented. While the first two topics had a publication as output, the last research was not closed at the end of the Project. Moreover, the tools were not ready for their inclusion on the MediFor evaluations, such that only internal experimental results are shown.

All these three research topics are described in this Final Performance Report. For each of them, we have a section divided into an introduction part, a methodology description, the analysis of the results and some final conclusions.

2 Content and meta-data based image integrity verification Introduction

Across the years, a great attention has been devoted to signal-based methods for image forgery detection with interesting results, even in automatic frameworks. Nevertheless, these methods are often ineffective when the investigated content undergoes a processing chain that may partially or completely spoil the traces left by previous operations. On the other hand, scene-based solutions can cope effortlessly with non-native contents, but they are not popular yet in the forensic domain, as they usually require specific features that are both difficult to detect and prone to noise, thus making it quite arduous to avoid altogether manual intervention. The first part of our research has been devoted to design new scene-based algorithms that show improved performance with respect to the state of the art. In particular, we worked on two problems: cropping detection, and face splicing detection.

Cropping is a simple yet powerful way to maliciously alter the content and the meaning of an image. Despite its communication impact, the forensic community has historically investigated this kind of forgery less than other image manipulations like splicing, copy-move or removal. The few solutions presented in literature are signal-based methods that look for blocking artefacts arising from image compression (Bruna, et al., 2011) (Li, et al., 2009). However, such solutions have problems dealing with images saved with a high-quality factor or after simple re-compression operations. For these reasons, during the MediFor project, we developed two different approaches for cropping detection: (i) a detector based on image meta-data analysis, and (ii) a method exploiting geometrical characteristic of the scene in order to estimate image projective invariants that can be used to assess the integrity of the image.

Concerning face splicing, this attack is achieved by inserting into an original image a human face retrieved from a different photo. This manipulation is one of the most critical since it deals with people’s identity and can be used to produce images where specific subjects are inserted into a particular and misleading context. During the Medifor Program we developed a novel technique to detect face splicing based on the physical analysis of the imaged scene. Previous works exploiting physical traces in the image try to directly extract and estimate the light parameters (i.e., the light source position, color and intensity) on each single face in the image and to detect inconsistencies indicating possible tampering. These parametric models that describe the interaction between light and environment are based on the spherical harmonics representation (Ramamoorthi & Hanrahan, 2001) (Basri & Jacobs, 2003). A possible splice in the image is noticed when, in the same image, light coefficients computed starting from different parts of the scene or objects exhibit relevant differences. In particular, light coefficients are estimated from occluding boundaries in (Johnson & Farid, 2007), and from human faces in (Peng, et al., 2015) (Peng, et al., 2016) (Peng, et al., 2017), after retrieving their 3D shape. To the best of our knowledge, the complex model described in (Peng, et al., 2017), enriched to overcome the strict assumption of the spherical harmonics representation, represents the current state-of-the-art in face splicing, but still shows the main drawbacks inherent in retrieving the spherical light coefficients. Differently, our new approach tries to indirectly analyze physical discrepancies as alterations measured on histograms that statistically model the interaction between a single face, with its own geometry and shape, and light.

2.1 Methodology

2.1.1 Meta-data based cropping detector

A naïve cropping attack can be spotted by checking some meta-data of an image, without requiring complex analysis on the image content. In Figure 1 the flow chart of the cropping detector is presented.

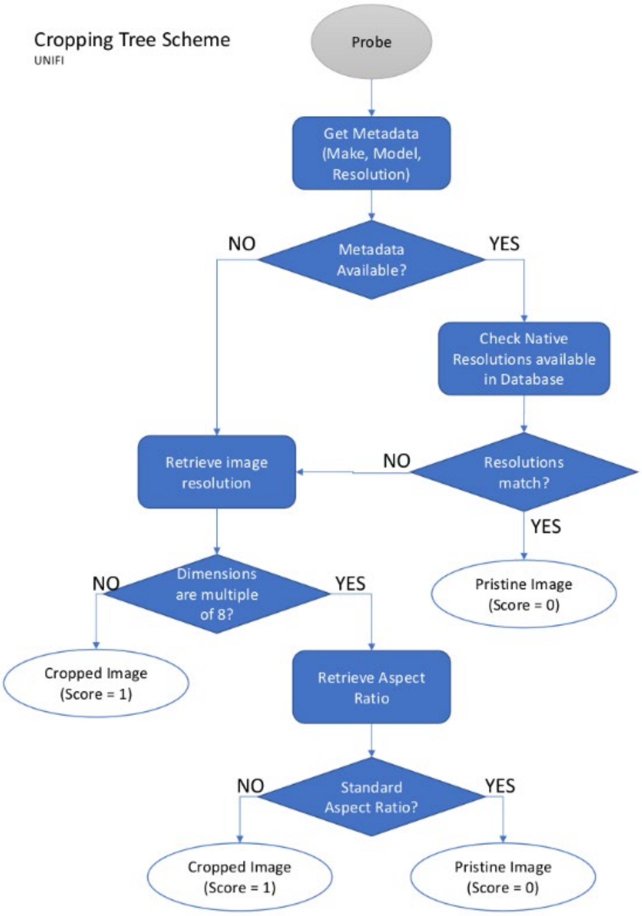

Cropping Tree Scheme
UNIFI
Probe
Get Metadata (Make, Model, Resolution)
Metadata Available?
NO
YES
Check Native Resolutions available in Database
Resolutions match?
NO
Retrieve image resolution
YES
Pristine Image (Score = 0)
Dimensions are multiple of 8?
NO
YES
Cropped Image (Score = 1)
Retrieve Aspect Ratio
Standard Aspect Ratio?
NO
YES
Cropped Image (Score = 1)
Pristine Image (Score = 0)

Figure 1: flow-chart of the meta-data based cropping detector.

As can be seen, at first the make, model and resolution of the probe image are extracted from the image meta-data. Then the system uses the retrieved make and model to extract standard resolution values from a built-in database that includes several camera models. In the case that the model is found in the dataset, if the standard resolution matches the probe resolution, the input image is labeled as pristine. On the other hand, if no meta-data are available in the probe, or if the probe model is not found in the dataset, or if there is no match between the database standard resolution and the probe resolution, the system performs two additional test. Firstly it checks if the image has dimensions divisible by 8 (since the JPEG compression works with 8x8 blocks): if the check fails, the probe is labeled as forged; otherwise, it verifies the image aspect ratio and labels the image as pristine or forged depending whether the probe has standard aspect ratio (e.g. 4/3, 16/9, etc.) or not.

2.1.2 Geometric-based cropping detector

This tool is based on the assumption that in modern cameras, the image principal point (PP) – i.e. the projection of the camera focal center to the image plane – falls near the image center (CC). By observing the position of the PP with respect to CC, we can assess if the image was pristine (PP coincides with CC) or cropped. Note that only asymmetric crops can be detected.

PP estimation is a known topic in computer vision and photogrammetry, strictly related to the camera calibration problem. When the camera is available, accurate off-line techniques exploiting a known pattern in the scene can be used to calibrate it (Zhang, 2000). The calibration problem can also be solved in the absence of the original camera, if images taken with that camera are available, in which case the problem is better known as self-calibration. Several self-calibration techniques exist, which differ according to the type of visual data (videos, image collections, single images) and operating conditions (e.g., in a video, fixed vs changing camera parameters) (Szeliski, 2010). Self-calibration of single images typically relies on a priori information about the scene structure, which can be exploited to infer the calibration parameters (Colombo, et al., 2006) (Guillou, et al., 2000). Structural information of special relevance to applications is that of Manhattan World scenes (Coughlan & Yuille, 1999), where it is assumed that the scene includes man-made structures like buildings, giving rise to sets of lines having mutually orthogonal directions in 3D (Deutscher, et al., 2002) (Pflugfelder & Bischof, 2005). These lines, once projected onto the image plane using a pinhole camera model, can be used to estimate the vanishing points of the scene. In case of Manhattan World scenes, most of the image lines are projection of mutually orthogonal 3D directions. Exploiting this knowledge, it is possible to estimate the intrinsic camera parameters, that also includes the location of image PP by solving a linear system.

Transferring to the forensic domain computer vision techniques, which typically assume genuine images, make the task of camera calibration (and specifically PP estimation) even more challenging – for example, PP is often initially assumed to be in the image center, and then either used as is or slightly refined. Conversely, in common forensic scenarios, only images of unknown origin are available, and no a priori assumptions can be made about parameters. This means that any parameter to be exploited for tampering detection must be extracted directly from (possibly manipulated) image data, without any prior information about it.

This tool is designed to detect evidence of cropping in a large collection of images. This requires that the algorithm operates in an automatic way, being also capable to decide autonomously whether the image at hand is tractable (i.e., it meets the Manhattan-world scene assumption) or not.

The algorithm is designed as follows. After detection of straight lines, these are clustered in order to estimate a set of three vanishing points related to mutually orthogonal directions in 3D. From them, vanishing points are computed and a first candidate PP is obtained. Evidence of cropping is then established with a statistical analysis of a cloud of putative PPs extracted from the image with a Monte Carlo process. Two heuristic criteria are then introduced to discard intractable images:

(i) *MaxAngle*: as reported in (Row & Reid, 2012), a triangle joining vanishing points related to three mutually orthogonal directions in the 3D space can't have angles greater than 90°. If a greater angle is found among the extracted VPs, the image is opted-out immediately, without wasting additional time on its analysis.

(ii) *MaxDist*: the distance between the ground truth PP and the cropped image center (normalized w.r.t. the diagonal of the cropped image) can be expressed as a function of the cropping factor $\alpha \in [0,1[$ as

$$\mathcal{S}(\alpha) = \frac{\alpha}{2(1-\alpha)}$$

Since we assumed to handle cropping factors up to 50%, with maximum expected distance equal to $\mathcal{S}(1/2) = 0.5$, the image at hand is opted-out without entering the Monte Carlo analysis if the first candidate PP distance w.r.t the CC exceeds 0.5.

In Figure 2, a graphical representation of the developed pipeline is reported. Further details can be found in our publication (Fanfani, et al., 2020).

Note also that a preliminary analysis on the reliability of PP estimation in the forensic scenarios was reported in our paper (Iuliani, et al., 2017).

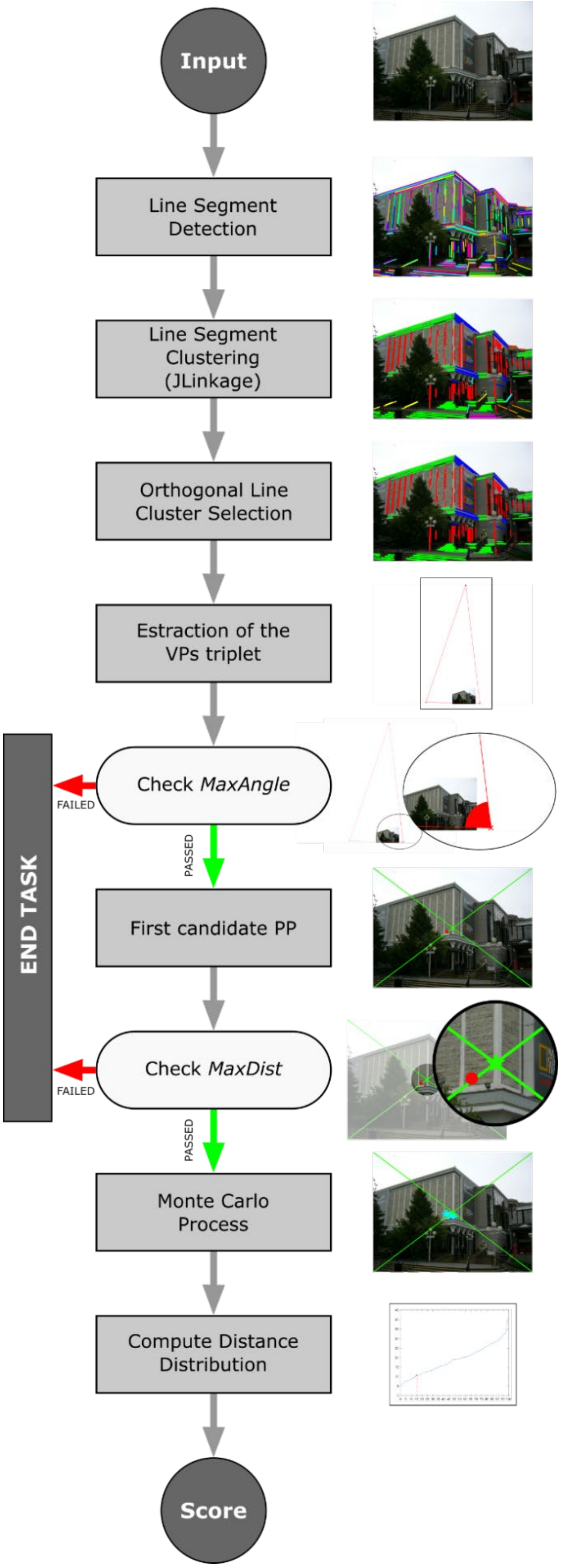


Figure 2: geometric cropping detector pipeline.

2.1.3 Face splicing detector by means of FISH descriptors

Under the assumption of convex and Lambertian surfaces with fixed albedo and distant light sources, the image intensity values of points in the scene only depend on their associated surface normals. In the case of faces, the resulting channel-wise mapping function $\mathrm{L}: \mathrm{R}^3 \rightarrow \mathrm{R}$ from normals $\mathrm{n} = [\mathrm{x\,y\,z}]^{\mathrm{T}}$, $\mathrm{z} > 0$ to a color channel intensity of the image $\mathrm{I} = \mathrm{L}(\mathrm{n})$ can be statistically modeled using a histogram-based representation, referred to as *Face Intensity-Shape Histogram* (FISH), computed as follows.

Given a face in the image and its associated 3D shape model, we first pre-process the model so as to remove face regions strongly violating the assumptions above. These areas include neck and ears (that yield poorly estimated normals), mouth, eyes and eyebrows (that have a different albedo and reflectance with respect to face skin), and saturated areas.

FISH bins $\mathrm{i} = 0, \dots, \mathrm{b}$ are sampled according to the vertexes of a semi-icosphere (i.e., the simplicial polyhedron at subdivision level 3 approximating a semi-sphere limited to the positive z-axis, $\mathrm{b} = 304$). Each bin corresponds to a distinct quantized surface normal $\mathrm{n_i}$. FISH bin values $\mathrm{I_i} = \mathrm{L}(\mathrm{n_i})$ for each color channel are computed via Gaussian kernel density estimation as explained hereafter. Let $\widehat{\mathrm{I_k}}$ and $\widehat{\mathrm{n_k}}$ be respectively the intensity value and the associated normal of a point on the masked face. Then:

$$\mathrm{I_i} = \sum_{\mathrm{k}} \frac{\mathrm{w_{ik}}}{\mathrm{w_i}} \widehat{\mathrm{I_k}}$$

where the sum is over the masked face pixels, with weights

$$\mathrm{w_i} = \sum_{\mathrm{k}} \mathrm{w_{ik}}$$

computed from the Gaussian distribution

$$\mathrm{z_{ik}} = \frac{1}{\sqrt{2\pi\sigma}} \mathrm{e}^{-\frac{1}{2}\left(\frac{\arccos(\mathrm{n_i} \cdot \widehat{\mathrm{n_k}})}{2\sigma}\right)^2}$$

subject to an influence cutoff threshold τ

$$\mathrm{w_{ik}} = \mathrm{z_{ik}} \text{ if } \frac{z_{ik}}{\sum_i z_{ik}} > \tau$$

or 0 otherwise.

The standard deviation σ used to define the kernel bandwidth is equal to the average angular distance between two adjacent vertexes of the icosphere.

By concatenating the bin values for each channel, the final FISH descriptor L is obtained. FISH descriptors can be used to compare faces in a probe image. The more two FISH descriptors are similar, the more the corresponding faces are likely to be exposed to the same light conditions.

A possible definition of the distance $\mathcal{D}(a, b)$ between two FISH descriptors L^a and L^b associated to faces *a* and *b* is

$$\mathcal{D}(a,b) = \left(\sum_{\substack{i=0,\ldots B \\ (w_i^a>0)\wedge(w_i^b>0)}} \left\| \boldsymbol{I}_i^a - \boldsymbol{I}_i^b \right\|^2 \right)^{\frac{1}{2}}$$

where $I_i^a = L^a(n_i)$, $I_i^b = L^b(n_i)$, $\|\cdot\|$ is the Euclidean norm---chosen experimentally, as it gives the best results---and w_i^a, w_i^b are the bin weights.

However, unhandled skin albedo would result in an incorrect FISH-based face matching. In order to remove skin color effects when comparing two FISH descriptors $\boldsymbol{L^a}$ and $\boldsymbol{L^b}$, we developed and tested two normalization strategies. The first strategy consists of simply pre-normalizing $\boldsymbol{L}$ by the mean RGB value $\boldsymbol{\mu}$ of the associated masked face, under the common assumption that albedo is a scale factor, i.e.

$$\dot{I}_\iota = \dot{L}(n_i) = L(n_i)/\mu$$

channel-wise, so that

$$\mathcal{D}'(a,b) = \mathcal{D}\left(\dot{L}^a, \dot{L}^b\right)$$

In the second strategy, the albedo characterizing $\boldsymbol{L^a}$ is replaced with that of $\boldsymbol{L^b}$ taking into account color saturation, i.e.,

$$I_i^{a\to b} = L^{a\to b}(n_i) = \min\left(255, L^a(n_i)\,\frac{\mu_b}{\mu_a}\right)$$

and vice-versa, so that

$$\mathcal{D}''(a,b) = \min\left(\mathcal{D}(L^a, L^{b\to a}), \mathcal{D}(L^b, L^{a\to b})\right)$$

2.1.3.1 Automatic face splicing detection

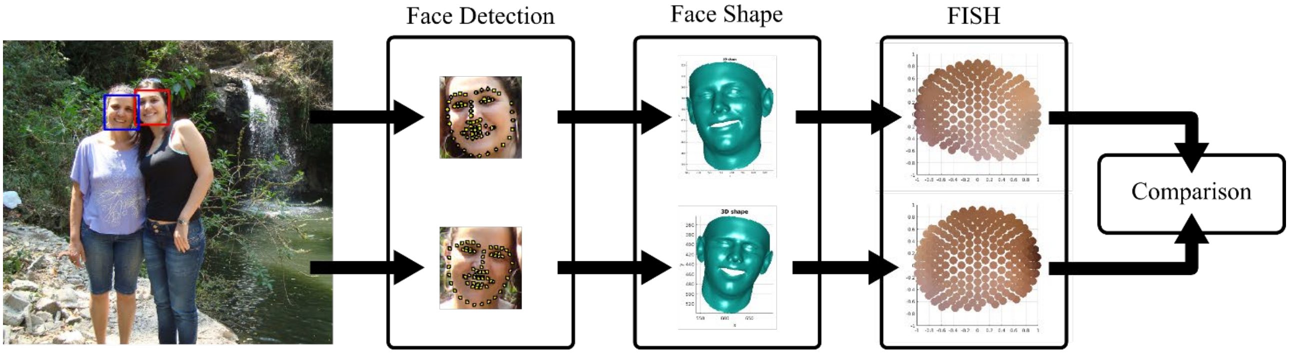


Figure 3: face splicing detection pipeline.

We employed the FISH descriptor to develop a fully automated pipeline for face splicing detection that can be divided into the following three steps (see Figure 3):

- **Face detection**. The method proposed in (Mathias, et al., 2014) is used, also registering on each recognized face using 68 landmarks according to the face alignment algorithm of (Xiong & la Torre, 2013).

- **Face shape and normals estimation**. Face landmarks computed at the previous step are used to register a *3D Morphable Model* (3DMM) and to obtain an estimate of the face shape. In particular, we adopted the solution presented in (Zhu, et al., 2015), combining the *Basel Face Model* (Paysan, et al., 2009) and the *Face Warehouse* model (Cao, et al., 2014) in order to be able to adapt the model to both identity and expression. As an alternative approach, we also tested the recent method proposed in (Trigeorgis, et al., 2017) based on convolutional neural networks.
- **FISH descriptors extraction and comparison**. As previously described.

Note that, in the case that only two faces are detected, the pipeline can detect the occurrence of tampering, but is unable to indicate which of the two is the tampered face, while, if more than two faces are found, the spliced face can be localized as the one with the greatest distance in terms of FISH descriptors from the other faces.

2.2 Results

2.2.1 Results for meta-data based cropping detector

We tested our meta-data based cropping detector on MFC19-EvalPart1-Image-Ver1, that is a part of the Media Forensics Challenge dataset built by the NIST for the MediFor Project and used as benchmark to evaluate the developed algorithms during the four project years, obtaining the following results: AUC = 0.59 in the case of no selective scoring, while, limiting to crop manipulation (i.e. Selective Scoring CROP), an AUC = 0.70 is obtained, not a bad result considering also that its running time is about 3 seconds per probe. Note that this tool cannot provide any localization of the tampering, neither does it have heuristics to opt-out probes.

2.2.2 Results for geometric-based cropping detector

We deeply tested our geometric-based cropping detector on several publicly available datasets including man-made scenes, showing good performance and high robustness w.r.t. re-compression, enhancement, and blurring operations. Please refer to our published paper (Fanfani, et al., 2020). Also, in the supplementary material, we reported additional plots of ROC curves.

On the other hand, this tool was also tested on MFC19-EvalPart1-Image-Ver1 obtaining AUC of 0.49 and 0.52 with no selective scoring, respectively for no opt-out and opt-out. Considering only cropping attack (i.e. selective scoring CROP), it achieved AUC of 0.49 and 0.51 in the case of no opt-out and with opt-out.

The different performance obtained in the public dataset with respect to those achieved on MFC evaluation, indicate how this method is strongly dependent on the input probe scene: when dealing with general kind of scenes in the MFC dataset, a drop in performance is obtained. Also, we can observe that the heuristic criteria used in the software, while effective in discarding wrongly estimated probes, are not sufficient to discriminate between tractable scenes (i.e. man-made scenes) and intractable ones.

2.2.3 Results for face splicing detection

Detailed results on publicly available datasets are reported in our journal paper (Fanfani, et al., 2019).

Hereafter in Table 1, we report results obtained on the MFC19-EvalPart1-Image-Ver1 dataset during MediFor evaluation. Note that, the method opted-out probes if less than two faces are found in the image, since in that case no comparison are possible.

Table 1: AUC values of face splicing detection results on MFC19.

	No OptOut	OptOut
No selective scoring	0.53	0.58
Selective scoring [FaceManip]	0.70	0.64

2.3 Conclusions

During the MediFor project, we developed two cropping detectors, one based on the analysis of image meta-data, the other based on geometric/projective constraints to detect cropping. Intermediate results were obtained: while the meta-data based detector can achieve interesting results on cropping attacks, this is strongly related to the quality of the manipulation: we think that an attacker with some experience and knowledge of the detector could easily fool this software. On the other hand, the geometric based detector offers a higher robustness to anti-forensic attack, and since it exploits a projective invariant of the image, it cannot be easily fooled. However, its application is strongly limited to particular kinds of scenes, and its use in-the-wild should be careful unless a high confidence scene classification (not available at this moment) is used to filter out intractable probes.

Moreover, we proposed a novel approach to face splicing detection based on light analysis. The novel FISH descriptor is designed according to a statistical representation based on histograms, implicitly mapping image intensities and 3D normal vectors. FISH can alleviate the impact of the low accuracy of the 3D face model, typically strongly affecting the methods based on spherical harmonics.

While the performance of the proposed methods under controlled scenarios, described in papers (Fanfani, et al., 2020) and (Fanfani, et al., 2019) are of interest, the results in the MediFor Evaluation have not been satisfying, such that these research paths have been discontinued.

3 Container-based video integrity verification

3.1 Introduction

Integrity verification of digital videos is still mostly an uncharted territory; there are, however, several studies regarding integrity verification of images, where the approach is to analyze the file format and metadata and determine their compatibility, completeness, and consistency with respect to the context in which it is assumed the resource has been created. More specifically, JPEG coding data, Exchangeable image file format (Exif) metadata and thumbnail size have been studied: since each acquisition device and processing software usually adopts customized quantization tables, it is possible to exploit these differences to address the source identification problem (Kee, et al., 2011). Subsequent studies revealed that the file structure, too, contains a lot of information about the history of a content, while being much more difficult to extract and modify for a user than metadata. Available editing software and metadata editors, in fact, do not have the functionality to modify such low-level information, like the internal order of the core file structures (Gloe, 2012).

These studies have been recently extended to digital videos too. In (Gloe, et al., 2014), Gloe et al. explore the low-level characteristics represented by metadata and low-level file format information, with an emphasis on the structure of the video file container. Indeed, video standards prescribe only a limited number of characteristics for the data container formats, thus leaving a lot of discretion to the manufacturer; this lead to differences that can be exploited for forensic purposes. However, while providing a pioneering exploration of video container formats from a forensic viewpoint, the manual approach proposed in (Gloe, et al., 2014) reduces the forensic capabilities on new generation media where the containers may be huge, deeply nested and strongly variable among different models; furthermore the discriminative power of some container features can be hardly quantified by manual inspection.

We designed a new approach (Iuliani, et al., 2018) for unsupervised analysis of video file containers by providing a formal metric to automatically quantify the dissimilarity between two containers. The measure accounts for both the container structure and content, and has proven to be effective in distinguishing videos whose integrity is preserved from videos whose integrity is compromised.

The method proposed in (Iuliani, et al., 2018) merely detects a loss of integrity, without providing a human-interpretable explanation of the reasoning behind its decisions. Then, we introduced a container-based method to identify the software used to perform a video manipulation. This is achieved by using a decision-tree-based classifier applied to a vectorial representation of the video container structure. Decision Trees (Quinlan, 1986) are a non-parametric learning method used for classification problems in many signal processing fields. Their key feature is the ability to break down a complex decision-making process into a collection of simpler decisions. The proposed method (Yang, et al., 2020), simply called EVA (Efficient Video Analysis), offers several forensic opportunities, such as: identifying the manipulating software (e.g. Adobe Premiere, ffmpeg, . . .); providing additional information related to the original content history, such as the source device operating system. The process is extremely efficient since a decision can be taken by checking the presence of a small number of features, independent of the video length or size. Furthermore, EVA can provide a simple explanation for the process leading to an outcome, since container symbols used to take a decision can be inspected.

These two methods are based on the analysis of the structure of the file containing the video. Indeed, most smartphones and compact cameras output videos in mp4, mov, or 3gp format. This video packaging refers to the same standard, ISO/IEC 14496 Part 12 (ISO/IEC 14496, 2008) that defines the main features of MP4 (ISO/IEC 14496, 2003) and MOV (Apple Computer, Inc., 2001), containers while leaving a wide margin for those who implement it. In Figure 4 we provide an example of an MP4-like container, a tree like structure describing the video file with respect to three aspects: how the bytes are organized (physical aspect); how the audio/video streams are synchronized (temporal aspect);and how the latter two aspects are linked (logical aspect). Each node (atom) is identified by a unique 4-byte code. It consists of a header which describes its role in the container and possibly some associated data. The first atom to appear in a container has to be ftyp, since it defines the best usage and compatibility of the video content.

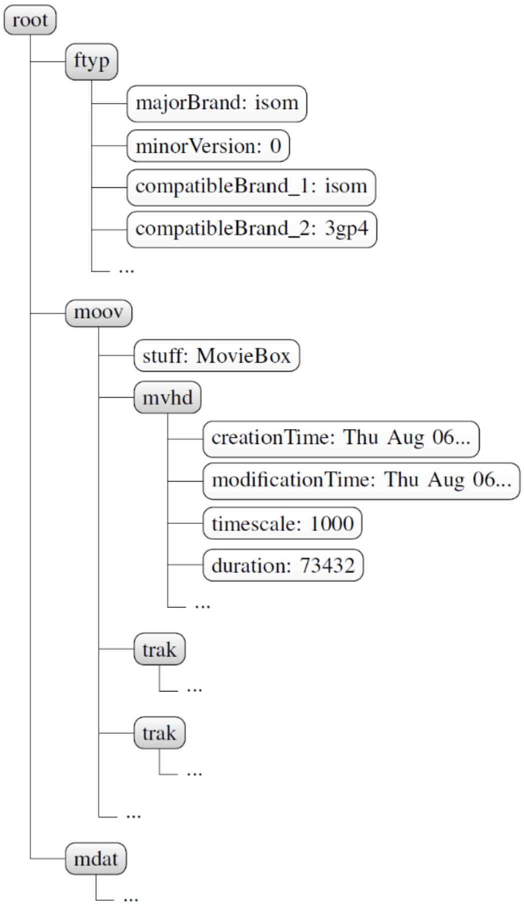


Figure 4: Video container example

We represent the video container as a labelled tree where internal nodes are labelled by atoms names (e.g. moov) and leaves are labelled by field-value attributes (e.g. @stuff: MovieBox). To take into account the order of the atoms, each XML-node is identified by a 4-byte code of the corresponding atom along with an index that represents the relative position with respect to the other siblings at a certain level.

3.2 Methodology

3.2.1 Video analysis based on container dissimilarities

Given a video X, its container is then represented as an ordered collection of atoms $a_1, \dots, a_n$, possibly nested. Each atom can be described as a set of field-value attributes $a_i = \{\omega_1(a_i), \dots, \omega_{m\_i}(a_i)\}$. By combining the two previous descriptions, the video container can be characterized by the set of field-value attributes $X = \{\omega_1, \dots, \omega_m\}$, each with its associated path $P_X(\omega_i)$ that is the ordered list of atoms to be crossed to reach ω_i in X starting from the root.

In summary, the video container structure is completely described by a list of m field-value attributes $X = \{\omega_1, \dots, \omega_m\}$, and their corresponding paths $\{P_X(\omega_1), \dots, P_X(\omega_m)\}$.

When a video is processed in any way, even without further encoding, the container structure is strongly altered with respect to its native structure. Conversely, the file containers of native content generated from a specific source device are expected to have a small intra-variability, caused by differences in the device settings. More generally, given a video X whose integrity has to be assessed, and a native reference video X' coming from the same supposed device model, their container structure dissimilarities can be exploited to expose evidences of integrity violation, as follows: we define their similarity $S(X, X')$ as the percentage of shared field-values with corresponding paths. Then, their dissimilarity can be computed as the mismatching percentage of all field-values, i.e., $mm(X, X') = 1 - S(X, X')$. To preserve symmetry we compute as final metric $D(X, X') = (mm(X, X') + mm(X', X))/2$. Technical details related to this computation are reported in (Iuliani, et al., 2018)

3.2.2 Video analysis based on decision trees

A video container X can be characterized by the set of symbols $\{s_1, \dots, s_m\}$, where s_i can be: (i) the path from the root to any field (value excluded), also called field-symbols; (ii) the path from the root to any field-value(value included), also called value-symbols.

An example of this representation can be:

```
s1 = [ftyp/@majorBrand]

s2 = [ftyp/@majorBrand/isom]

…

si = [moov/mvhd/@duration]

si+1 = [moov/mvhd/@duration/73432]
```

Overall, we denote with Ω the set of all unique symbols $s_1, \dots, s_M$ available in the world set of digital video containers $\mathrm{X} = \{X_1, \dots, X_N\}$. Similarly, $\Gamma = \{C_1, \dots, C_S\}$ denotes a set of possible origins (e.g., Huawei P9, Apple iPhone 6s). Given a container X, the different structure of its symbols $\{s_1, \dots, s_m\}$ can be exploited to assign the video to a specific class C_u. For this purpose binary decision trees (Safavian & Landgrebe, 1991) are employed to build a set of hierarchical decisions. In each internal tree node, the input data is tested against a specific condition; the test outcome is used to select a child as the next step in the decision process. More specifically, in our approach we adopted the growing-pruning-based Classification And Regression Trees (CART) (Breiman, 2017). Given the size of unique symbols $|\Omega| = \mathrm{M}$, a video container X is converted into a vector of integers $\mathrm{X} = \{x_1, \dots, x_M\}$

where x_i is the number of times that s_i occurs into X. This approach is inspired by the bag-of-words representation (Schütze, et al., 2008) used to reduce variable-length documents to a fixed-length vectorial representation. Note that X contains several symbols that are not representative of any class, thus contributing to class intra-variability only (e.g. information related to video length, acquisition date and time). These symbols are useless to determine the source of a video and they should be possibly removed. Thus, we pre-filtered the data as explained in (Yang, et al., 2020).

3.3 Results

3.3.1 Video analysis based on container dissimilarities

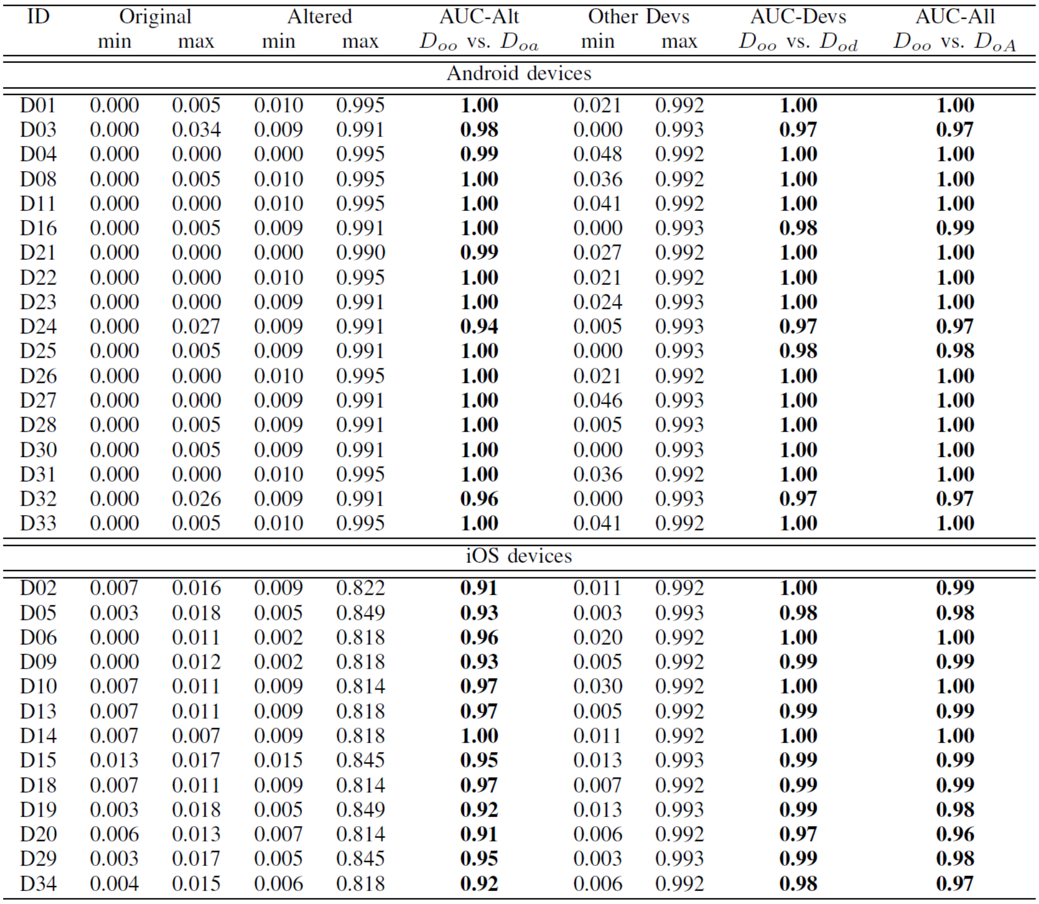
We tested the proposed techniques on the VISION dataset (Shullani, et al., 2017), analyzing 31 portable devices of 8 major brands that leads to an available collection of 578 videos in the native format plus their corresponding social versions (YouTube and WhatsApp are considered).

We considered four different scenarios of integrity violation:

- WhatsApp: the video is exchanged through the WhatsApp social platform, that performs a strong modification of both the data stream and file container structure (the video is re-encoded and possibly downscaled);
- YouTube: the video is exchanged through the YouTube. The videos were uploaded via the YouTube web interface and downloaded at the maximum resolution available with ClipGrab.
- FFmpeg: the video is cut after 10 seconds using FFmpeg, but without re-encoding the stream;
- ExifTool: only datetime-related metadata are edited using ExifTool.

For each of the four cases, we adopted the following procedure: we considered the set of videos $X_1, \dots, X_{N_i}$ available for each device C_i, and we computed the intra-class dissimilarities between two native videos $D_{ij} = D(X_i, X_j)\ \forall i \neq j$. For simplicity we denote with D_{oo} this set of dissimilarities. Then, we considered the corresponding inter-class dissimilarities $D_{i,j}^t = D(X_i, X_j^t)\ \forall i \neq j$ between a native video X_i and the corrupted version X_j^t obtained with the tool t (WhatsApp, YouTube, FFmpeg, or ExifTool) applied to X_j . We denote with D_{oa}^t this set of dissimilarities.

By applying this procedure to all the considered devices, we collected 2890 samples for both D_{oo} and any of the four D_{oa}^t. The results of this test are reported in Figure 5: the first column (ID) indicates the unique number identifying the device according to the VISION Dataset nomenclature, the "Original" column shows the minimum and maximum values of the intra-class dissimilarity D_{oo} obtained for each device; the "Altered" column reports the minimum and maximum values of the inter-class statistics D_{oa}^t for all the considered tools (WhatsApp, YouTube, FFmpeg, or ExifTool); the "AUC-Alt" column shows the Area Under Curve summarizing the performance of the proposed method in distinguishing original and altered videos: as it clearly appears from the table, in most of the devices the maximum value of D_{oo} is lower than the minimum value of D_{oa}^t, thus indicating that the two classes can be separated perfectly; this corresponds to the value 1 of the AUC; for some devices this does not hold, but in every case the AUC is very close to 1, indicating that the errors are limited.



ID	Original		Altered		AUC-Alt	Other Devs		AUC-Devs	AUC-All
	min	max	min	max	D_{oo} vs. D_{oa}	min	max	D_{oo} vs. D_{od}	D_{oo} vs. D_{oA}
Android devices									
D01	0.000	0.005	0.010	0.995	**1.00**	0.021	0.992	**1.00**	**1.00**
D03	0.000	0.034	0.009	0.991	**0.98**	0.000	0.993	**0.97**	**0.97**
D04	0.000	0.000	0.000	0.995	**0.99**	0.048	0.992	**1.00**	**1.00**
D08	0.000	0.005	0.010	0.995	**1.00**	0.036	0.992	**1.00**	**1.00**
D11	0.000	0.000	0.010	0.995	**1.00**	0.041	0.992	**1.00**	**1.00**
D16	0.000	0.005	0.009	0.991	**1.00**	0.000	0.993	**0.98**	**0.99**
D21	0.000	0.000	0.000	0.990	**0.99**	0.027	0.992	**1.00**	**1.00**
D22	0.000	0.000	0.010	0.995	**1.00**	0.021	0.992	**1.00**	**1.00**
D23	0.000	0.000	0.009	0.991	**1.00**	0.024	0.993	**1.00**	**1.00**
D24	0.000	0.027	0.009	0.991	**0.94**	0.005	0.993	**0.97**	**0.97**
D25	0.000	0.005	0.009	0.991	**1.00**	0.000	0.993	**0.98**	**0.98**
D26	0.000	0.000	0.010	0.995	**1.00**	0.021	0.992	**1.00**	**1.00**
D27	0.000	0.000	0.009	0.991	**1.00**	0.046	0.993	**1.00**	**1.00**
D28	0.000	0.005	0.009	0.991	**1.00**	0.005	0.993	**1.00**	**1.00**
D30	0.000	0.005	0.009	0.991	**1.00**	0.000	0.993	**1.00**	**1.00**
D31	0.000	0.000	0.010	0.995	**1.00**	0.036	0.992	**1.00**	**1.00**
D32	0.000	0.026	0.009	0.991	**0.96**	0.000	0.993	**0.97**	**0.97**
D33	0.000	0.005	0.010	0.995	**1.00**	0.041	0.992	**1.00**	**1.00**
iOS devices									
D02	0.007	0.016	0.009	0.822	**0.91**	0.011	0.992	**1.00**	**0.99**
D05	0.003	0.018	0.005	0.849	**0.93**	0.003	0.993	**0.98**	**0.98**
D06	0.000	0.011	0.002	0.818	**0.96**	0.020	0.992	**1.00**	**1.00**
D09	0.000	0.012	0.002	0.818	**0.93**	0.005	0.992	**0.99**	**0.99**
D10	0.007	0.011	0.009	0.814	**0.97**	0.030	0.992	**1.00**	**1.00**
D13	0.007	0.011	0.009	0.818	**0.97**	0.005	0.992	**0.99**	**0.99**
D14	0.007	0.007	0.009	0.818	**1.00**	0.011	0.992	**1.00**	**1.00**
D15	0.013	0.017	0.015	0.845	**0.95**	0.013	0.993	**0.99**	**0.99**
D18	0.007	0.011	0.009	0.814	**0.97**	0.007	0.992	**0.99**	**0.99**
D19	0.003	0.018	0.005	0.849	**0.92**	0.013	0.993	**0.99**	**0.98**
D20	0.006	0.013	0.007	0.814	**0.91**	0.006	0.992	**0.97**	**0.96**
D29	0.003	0.017	0.005	0.845	**0.95**	0.003	0.993	**0.99**	**0.98**
D34	0.004	0.015	0.006	0.818	**0.92**	0.006	0.992	**0.98**	**0.97**

Figure 5: Video Integrity Verification Performance

In a second test, we showed that we can correctly identify when a video, although native, does not belong to a specific camera model. To this end, we built the inter-class dissimilarities $D\left(X_i, X_j\ \right) \forall i \neq j$, where X_i and X_j are native videos belonging to different camera models. We denote with D_{od} this set of dissimilarities. We report the results of this test in Figure 5: in the “Other Devs” column the minimum and maximum of the inter-class statistics D_{od}; in the “AUC-Devs” column the AUC summarizing the performance in distinguishing the two classes D_{oo} and D_{od}. Results highlight that the adopted measure is very good in separating a query native video from native videos belonging to other devices, since AUC values are always higher than 0.96.

Eventually, in the last column “AUC-All” we reported the AUC summarizing the performance in distinguishing between D_{oo} and all inter-class statistics $D_{oA} = D_{oa} \cup D_{od}$, so we have joined the two previous cases.

Summarizing, for the video integrity verification tests we obtained perfect discrimination for videos altered by social network or FFmpeg, while for ExifTool we obtain an AUC greater than 0.82 on 70% of the considered devices. It can be noted that, using state of the art methods based on data stream analysis, it would be nearly impossible to detect cutting with FFmpeg, since no re-encoding occurs; still, cutting an arbitrary portion of the video can be considered a realistic and powerful attack.

We highlight that the comparison between a video reference and a video query requires on average just 0.15 seconds (the computational cost has been computed on an Intel(R) Core(TM) i7-3770 CPU at 3.40GHz, running all algorithms by means of Python 2.7).

3.3.1.1 MediFor Evaluation

Finally, during the MediFor evaluation, the proposed method achieved an AUC of 0.98 and 0.99 on the MFC19 and MFC20 datasets respectively (the tool is tagged as unifi-ed209 in Figure 6).

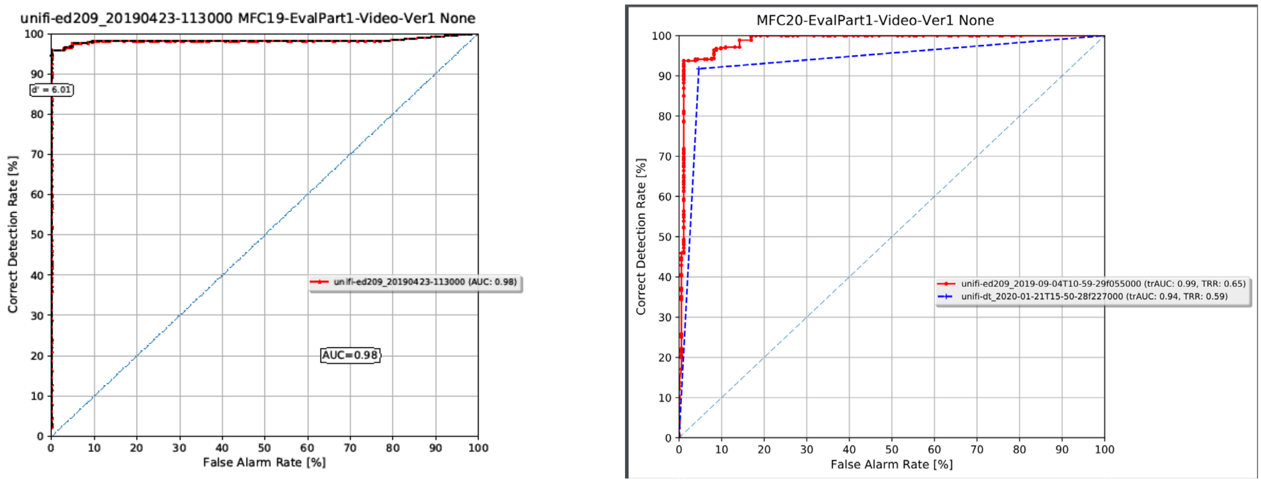


Figure 6: Performance on MediFor evaluation

Note that, with the latest version of the tool, we were able to process a video, independently on its length and size, within few tens of seconds (see Table 2).

Table 2: Performance of video container analysis on MFC 2020

File Format	#Videos	Ave Time (s)	Max Time (s)
.mp4	514	12.40	25.61
.mov	128	17.16	167.55
.3gp	41	13.57	15.37
.avi	27	12.23	87.30
.mts	12	N/A	N/A
.mxf	18	N/A	N/A

3.3.2 Video analysis based on decision trees

We performed tests under several scenarios:

- Integrity verification and comparison with other state of the art methods;
- Manipulation characterization and brand identification;
- Integrity verification on social media contents;
- Blind scenario where no hypothesis is made on the investigated content
- MediFor evaluation on MFC20

3.3.2.1 Integrity Verification

The first relevant experimental question is whether the proposed approach is capable of distinguishing between pristine and tampered videos. To answer it, we created a new collection of videos, starting from the VISION dataset (Shullani, et al., 2017), that includes native videos from 35 smartphones of 11 different brands. As it would have not been feasible to perform the editing operations, upload, and download of all the videos in VISION, we selected 4 videos for each device, thus obtaining a total of 140 pristine videos. Then, we created 1260 (140 x 9 editing operations) tampered videos, both automatically generated with ffmpeg and Exiftool, and manually created through Kdenlive, Avidemux and Adobe Premiere: Furthermore, all the produced contents (140 pristine videos and 1260 tampered ones) were exchanged through various social media platforms. Technical details related to the contents generation are available in (Yang, et al., 2020).

The container structure is extracted from each video by means of the MP4 Parser library (Apache, s.d.). Note that, due to how the dataset was built, some value-symbols are always present in some classes even if they are not relevant for their identification. For instance, all the cut videos have the same duration even if this is not, per se, relevant for identifying the editing. As this could lead to artificially higher performance, we manually removed the value-symbols associated to some fields as detailed in (Yang, et al., 2020).

In order to estimate the real-world performance of the proposed method we adopted an exhaustive leave-one-out cross-validation strategy. We partitioned our dataset in 34 subsets, each one of them containing pristine, manipulated, and social-exchanged videos belonging to a specific device. We performed each of the experiments hereby described 34 times, each time keeping one of the subsets out as test set, and using the remaining 33 for training our model. In this way, test accuracies collected after each iteration are computed on videos belonging to an unseen device. We reported the mean accuracies obtained among all the iterations as confusion matrices. During the training we assigned to each class a weight inversely proportional to the class frequency. We used the decision trees algorithms available as part of scikit-learn (Pedregosa, et al., 2011), a freely available Python toolkit for machine learning. We trained our method to distinguish between the two classes "Pristine" (containing 136 videos) and "Tampered"(containing 1224 videos). We obtained a global balanced accuracy of 98.5%, failing only for videos produced by D12 (Table 2).

We also compared our method with two recently proposed algorithms for video integrity (Iuliani, et al., 2018), (Güera, et al., 2019). In Table 3 we report the mean global accuracy and the average runtime per fold for the proposed approach and for those two methods. EVA outperforms the approach proposed by Guera et al. both in effectiveness and efficiency. When compared with Iuliani et al., EVA shows slighter better performance but a much faster computing time. Indeed, the cost for a decision tree analysis is O(1) since the output is reached in a constant number of steps; on the contrary, in Iuliani et al. O(N) comparisons are required since all the N reference set examples must be compared with a tested video. Furthermore, EVA often provides a simple explanation for the outcome.

Table 3: Performance comparison

	Balanced Accuracy	**Training Time (sec.)**	**Testing Time (sec.)**
Guera et al.	0.67	347	<1
Iuliani et al.	0.85	N/A	8
EVA	0.98	31	<1

3.3.2.2 Manipulation Characterization

We also performed a set of experiments designed to show that the proposed method, as opposed to the state of the art, is also capable of identifying the manipulating software and then we tried to answer the following questions:

A. Software identification: Is the proposed method capable of identifying the software used to manipulate a video? If yes, is it possible to identify the operating system of the original video?

B. Integrity Verification on Social Media: Given a video from a social media platform (YouTube, Facebook, TikTok or WeiBo), can we determine whether the original video was pristine or tampered?
C. Blind scenario: Given a video that may or may not have been exchanged through a social media platform, is it possible to retrieve some information on the video origin?

3.3.2.3 Software identification

In this scenario we only analyze videos that either are native, or that have undergone a manipulation. This time, however, we trained our algorithm to classify which software has been used to tamper the video, if any. Our classes are thus: "native" (136 videos), "Avidemux" (136 videos), "Exiftool"(136 videos), "ffmpeg" (680 videos), "Kdenlive" (136 videos),and "Premiere" (136 videos). In this experiment EVA obtained a global balanced accuracy of 97:6%; the detailed results, reported in Table 4, show that the algorithm achieved a slightly lower accuracy in identifying ffmpeg with respect to the other tools. This is reasonably due to the fact that ffmpeg library is used by other software and, internally, by Android devices.

Table 4: Integrity verification performance

	Native	Avidemux	Exiftool	ffmpeg	Kdenlive	Premiere
Native	0.97		0.03			
Avidemux		1.00				
Exiftool	0.01		0.99			
ffmpeg		0.01		0.90	0.09	
Kdenlive					1.00	
Premiere						1.00

3.3.2.4 Integrity Verification on Social Media

In this scenario we tested YouTube, Facebook, TikTok and Weibo videos to determine whether they were pristine or manipulated prior the upload. A summary of the results obtained by our method is reported in Table 5. We achieved global balanced accuracies of 0.76, 0.80, 0.79, and 0.60 on Facebook, TikTok, Weibo, and Youtube, respectively (see Table 5).

Table 5: EVA performance on social media contents

	Accuracy	TNR	TPR
Facebook	0.76	0.40	0.86
TikTok	0.80	0.51	0.75
Weibo	0.79	0.45	0.82
Youtube	0.60	0.36	0.74

Such results are characterized by low true negative rates, and thus it cannot be considered effective in this scenario, as many tampered videos are incorrectly classified as pristine. The poor performance is mainly due to the social media transcoding process that flattens the containers almost independently of the video origin. As an example, after YouTube transcoding, videos produced by Avidemux and by Exiftool have exactly the same container representation. We do not know how the videos are processed by the considered platforms due to the lack of public documentation, but we can assume that uploaded videos undergo custom/multiple processing. Indeed, social media videos need to be viewable on a great range of platforms, and thus need to be transcoded to multiple video codecs and adapted for multiple resolution and bitrate. Thus, it seems plausible that those operations could discard most of the original container structure.

3.3.2.5 Blind scenario

In this scenario we considered videos that may or may not have been exchanged through a social media platform and we would like to extract the most complete information possible. We used all the videos in our dataset and we trained our classifier to distinguish (i) whether the video was downloaded from a social media platform; (ii) whether the video has been tampered and, if so, which software has been used; (iii) whether the original video belong to an Android or iOS device. As a summary we found no performance decay with respect to the previous scenarios. Even without any prior knowledge of the video origin, we are still able to distinguish between native and tampered videos. Our method is also able of correctly identifying videos belonging to YouTube, Facebook, TikTok and Weibo, even though in this case it is not possible to make further claims on the video authenticity. Detailed results are reported in (Yang, et al., 2020).

3.3.2.6 MediFor Evaluation

EVA was tested during the MFC20 evaluation. In Figure 7 we report the AUC achieved on MFC20 (the method is labeled as unifi_dt). We obtained an area of 0.94. Note that we have a single point within the AUC since the method does not provide a soft score. When compared to the baseline method (labeled as unifi), EVA achieved slightly lower performance. However, it is extremely efficient since it performs analysis in less than a second (see Table 6).

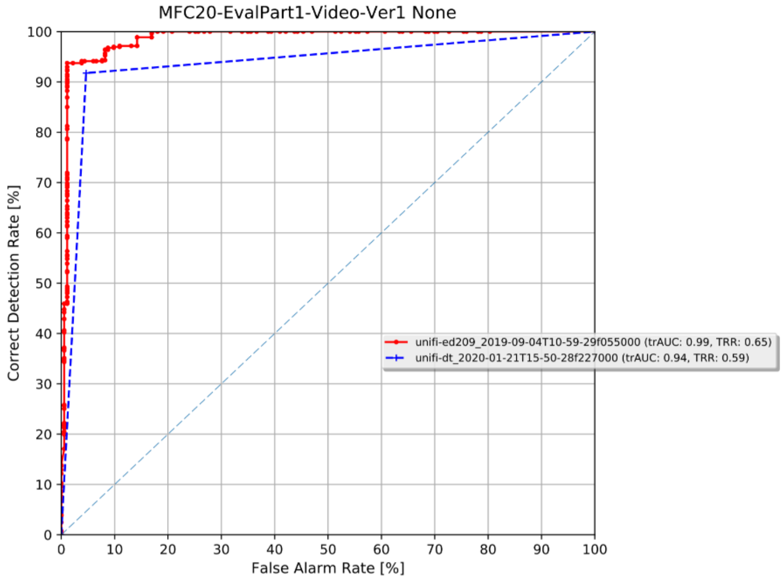


Figure 7: Performance of EVA on MFC20

Table 6: EVA performance on MFC20

File Format	#Videos	Ave Time (s)	Max Time (s)
.mp4	514	0.39	0.55
.mov	128	0.39	0.55
.3gp	41	12.70	13.85

3.4 Conclusions

We introduced two novel techniques for unsupervised forensic analysis of video file containers. The core idea is to exploit the differences in the file container structure and content introduced by different manufacturers, models and software processing.

Extensive experiments were carried out on publicly available datasets and during MediFor evaluations, showing excellent results for the integrity verification task. The proposed techniques are shown to be able to also automatically detect manipulations that are performed without video re-encoding, which is an unprecedented achievement for a video forensic algorithm. Moreover, the proposed approaches exhibit an extremely small computational cost as opposed to existing techniques based on video stream analysis. Moreover, the second proposed algorithm, in case of tampered videos, is able to characterize the software that performed the manipulation with an accuracy of 97.6%, even when the video is cut without re-encoding. As opposed to the state of the art, the proposed method is extremely efficient and can provide a simple explanation for its decisions. A new experimental dataset of 7000 videos was also created and shared with the research community, including contents generated with five editing tools and four social media platforms. The current limitation of the method is that a container based approach can identify whether the video belongs to a social medial platform, but it cannot be effectively applied on such contents for authenticity assessment, since the transcoding operation wipes out most of the forensic traces from the video container.

We achieved the highest AUC scores in the video manipulations task of MediFor evaluations, while at the same time being among the least computationally expensive algorithms.

4 PRNU analysis for smartphone identification

4.1 Introduction

This section presents our studies on PRNU for smartphone identification. In particular, this section includes four main topics: (i) a method for hybrid camera identification, that uses PRNU from images and videos; (ii) a calibration technique that can be reliably used to estimate the scale factors relating different acquisition modalities (image, video, and stabilized video) of a given device and a laboratory setup to deeply understand Electronic Image Stabilization (EIS). Then, (iii) a method exploiting deep neural networks to register PRNU signal under small scale and rotation transformations is presented.

Photo Response Non-Uniformity (PRNU) (Lukas, et al., 2006) is a unique fixed pattern noise generated during the acquisition process by any digital sensor. This makes PRNU ideal to develop effective methods for image and video source attribution (Chen, et al., 2008).

PRNU pattern is extracted pixelwise in order to derive the fingerprint of a device, implying that PRNUs are best generated and compared at native camera resolution (Shullani, et al., 2017). Due to their high sensitivity to pixel misalignments, PRNU patterns become particularly difficult to compare when the source images have been warped as result of the acquisition post-processing: this could happen for example when comparing images and videos of a same devices, that typically are obtained with different sensor portions and at different resolution. Moreover, the Electronic Image Stabilization (EIS) used in some devices to reduce shaking effect on recorded videos can further desynchronize the video PRNU w.r.t. the reference fingerprint obtained from flat-field images at full sensor resolution. More generally, we can state that even small scale and rotation transformations, introduced directly by the device or maliciously made by a forger, spoil the camera identification, strongly reducing correlation between the reference and probe PRNUs.

Current approaches need to tackle separately images and videos, or attempt to find an accurate estimate of the PRNU transformation as the one that maximizes the correlation in terms of Peak-to-Correlation-Energy (PCE) (Goljan & Fridrich, 2008), typically by brute-force search (Taspinar, et al., 2016) that is computationally expensive, and not always sufficiently accurate.

In the following we report the proposed methods to improve PRNU based source identification under several challenging scenarios.

4.2 Methodology

Hereafter we present each of the introduced topics. We devoted a different section to each topic, in order to ease the reading and provide better-structured presentation.

4.2.1 Hybrid reference-based Source Identification (HSI)

In the PRNU based source identification, image and video sources are still treated separately from one another. This approach is limited and anachronistic, if we consider that most of the visual media are today acquired using smartphones that capture both images and videos. We overcome this limitation by exploring a new approach that synergistically exploits images and videos to study the device from which they both come. Indeed, we prove it is possible to identify the source of a digital video by exploiting a reference sensor pattern noise generated from still images taken by the same device. To this aim, the geometrical relation between image and video acquisition processes are studied for 18 modern smartphones, including devices featuring in-camera digital stabilization. We also prove that the proposed technique, while preserving the state of the art performance for non-stabilized videos, is able to effectively detect the source of in-camera digitally stabilized videos as well. Most relevant related works can be found in (Iuliani, 2019).

In order to determine the source of a digital video (DV) based on a reference derived from still images we employed a two steps strategy: (i) the reference fingerprint is derived from still images acquired by the source device; (ii) the query fingerprint is estimated from the investigated video and then compared with the reference to verify the possible match. Moreover, the camera fingerprint K can be estimated from N still images (or frames) $I_1, \ldots, I_N$, captured by the source device. A denoising filter (Lukas, et al., 2006), (Mihcak, et al., 1999) is applied to each frame and the noise residuals $W_1, \ldots, W_N$ are obtained as the difference between each frame and its denoised version. Then the fingerprint estimate K is derived by the maximum likelihood estimator (Chen, et al., 2008). The fingerprint of the video query is estimated in the same way from the available videoframes. We will refer to these fingerprints as K_I and K_V respectively.

If the two fingerprints belong to the same camera sensor, we expect the existence of a 2D isometric transformation that maps the elements of K_I to those of K_V. Given a properly rescaled fingerprint, the source identification can be formulated as a two-channel hypothesis testing problem (Iuliani, 2019). The two-dimensional normalized cross-correlation $r(s_1, s_2)$ is calculated for each of the possible spatial shifts determined within a set of feasible cropping parameters. Then, given the maximum peak, its sharpness is measured by the PCE ratio (Goljan, et al., 2009). In order to consider the possible different scaling factors of the two fingerprints —since videos are usually resized— a brute force search can be conducted considering the PCE as a function of all plausible scaling factors. Then its maximum is used to determine whether the two fingerprints belong to the same device. Practically, if the maximum overcomes a threshold τ, the hypothesis that the two media belong to the same camera is decided and the corresponding peaks are exploited to determine the cropping and the scaling factors. It's worth noticing that recent devices feature electronic image stabilization as a means to reduce the impact of shaky hands on captured videos. Source identification of videos captured with active digital stabilization cannot be accomplished using the classical approach of PRNU fingerprint computation, since it would require the fingerprint to remain spatially aligned across all frames. This condition is not met due to the stabilization process (Mandelli, et al., 2019). HSI solves the problem on the reference side (the fingerprint is estimated from still images) but the issue remains on the query side. A first way to compensate digital stabilization was proposed in (Hoglund, et al., 2011) and tested on a single Sony device. Recently, in (Taspinar, et al., 2016), it was proposed to compute the fingerprint from a stabilized video by using the first frame noise pattern as reference, and registering all following frames on such reference by estimating the similarity transformation that maximizes the correlation between the patterns. The technique was proved to compensate for digital stabilization applied off camera by third party software with limited reliability, probably because the reference for the whole process is computed from a single frame. In the HSI paradigm, however, still images are exploited to estimate a more reliable fingerprint, while on the query side each video frame is registered on the image reference based on a similarity transformation.

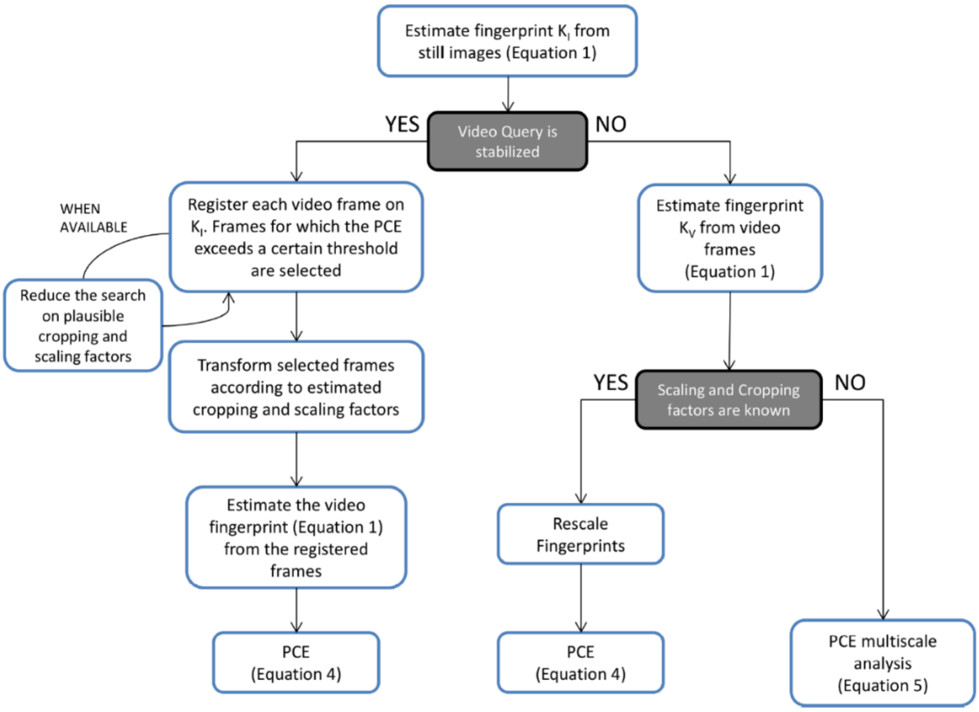

Estimate fingerprint K$_I$ from still images (Equation 1)
YES
Video Query is stabilized
NO
WHEN AVAILABLE
Register each video frame on K$_I$. Frames for which the PCE exceeds a certain threshold are selected
Estimate fingerprint K$_V$ from video frames (Equation 1)
Reduce the search on plausible cropping and scaling factors
Transform selected frames according to estimated cropping and scaling factors
YES
Scaling and Cropping factors are known
NO
Estimate the video fingerprint (Equation 1) from the registered frames
Rescale Fingerprints
PCE (Equation 4)
PCE (Equation 4)
PCE multiscale analysis (Equation 5)

Figure 8: Hybrid reference-based Source Identification (HSI) pipeline to source attribution of a query video.

Given a query video and a set of images belonging to a reference device, the proposed pipeline can be summarized in Figure 8. First, the device fingerprint K_I is estimated from still images. Then, stabilized videos are preliminarily identified by splitting the frames in two groups that are used independently to estimate two different fingerprints, as described in (Taspinar, et al., 2016), and by computing their PCE; a low PCE value will expose the presence of digital stabilization. If no stabilization is detected, the video fingerprint K_V is estimated by treating video frames as still images. Conversely, each frame is registered on the reference K_I searching for plausible parameters based on PCE values. In case the expected range of parameters is known, the search can be reduced to save computational effort and mitigate the false alarm probability. Only registered video frames for which the PCE exceeds a threshold t are then aggregated to estimate the video fingerprint K_V. Once both fingerprints K_I and K_V are available, their PCE is computed and the correlation value is compared to a threshold.

4.2.2 Understanding Electronic Image Stabilization

As the first step, the proposed approach requires to hold a device under investigation still on a static scene in order to acquire images from the different photo or video formats. The native full resolution photo serves as reference for the sensor grid, on which other image formats must be mapped into. It can be argued that this acquisition step would be not practicable in many application scenarios, since the procedure should be repeated for each specific device. Nevertheless, as verified later in the experimental section, the underlying PRNU pattern transformation depends only on the device model and not on the device exemplar at hand. This implies that, once estimated for one device, the same transformation holds for any other device of the same model. Figure 9 shows an example of the above acquisition step. In order to improve the registration accuracy, the scene must be in focus and include discriminative patterns distributed across the whole image area. In the case of videos, only I-frames are considered and, when available, acquired images are taken using remote or vocal controls in order to avoid any accidental misalignments due to camera shakes.

To register an output format to the reference, corner-like keypoints are extracted with the HarrisZ detector (Bellavia, et al., 2011) and matched with the SIFT-like sGLOH2 local image descriptor (Bellavia & Colombo, 2018). Given the initial set of correspondences, RANndom SAmple Consensus (RANSAC) (Fischler & Bolles, 1981) robustly models the warping – the model assumes scale and translation changes only. The scale factor is the most important parameter and it is fixed for any device output format, even in case of EIS. In case EIS is *off* translation is also fixed, while in case EIS is *on* it can be easily recovered by PCE maximization when EIS does not involve frame rotations. An example of registration is shown in Figure 9, notice that video frames cover a smaller portion when EIS is on in order to compensate for translations and rotations while avoiding missing image spots from areas not covered by the camera sensor.

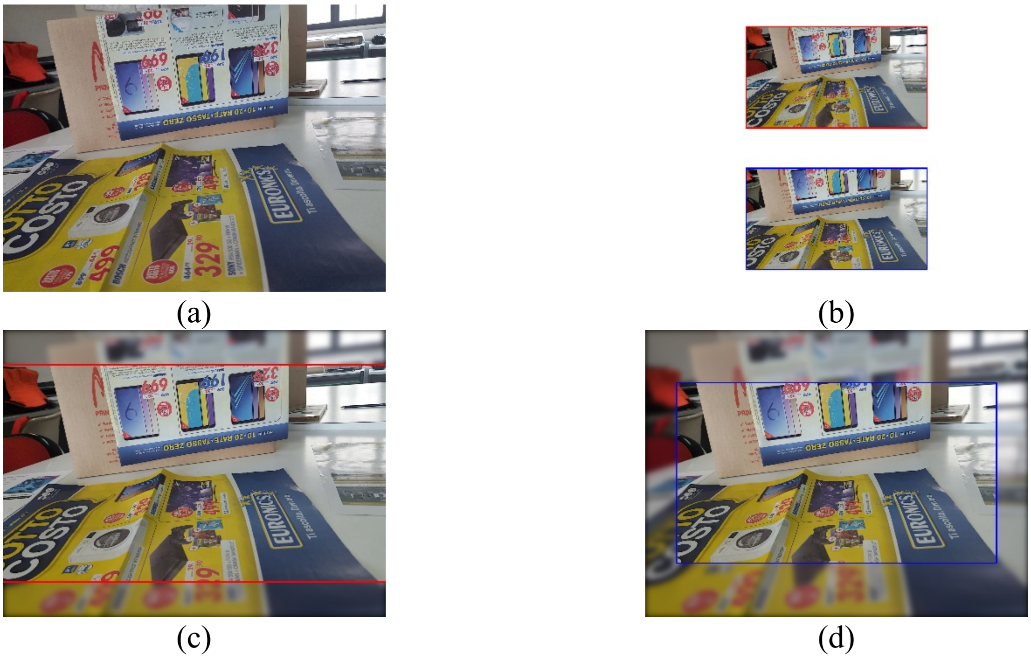

(a) (b) (c) (d)

Figure 9: Static scene image registration on a Samsung Galaxy S7 smartphone. The native full resolution photo (a) is used as reference to register the corresponding video frame in case EIS off (b, top) and on (b, bottom) using image keypoint matching. The final aligned video frames superimposed on the reference image are shown in (c) and (d), respectively. All images are scaled according to their resolution. The reference image on (c) and (d) is blurred for a better visual comparison.

RANSAC estimation requires setting the inlier reprojection error, indirectly setting the degree of uncertainty in the final model. According to this observation, the transformation estimated so far can be refined through an exhaustive search over a small set of allowable scales, translations, and rotations, operating analogously to other PRNU pattern alignment approaches. Specifically, the PRNU correlation in terms of PCE is evaluated over a limited set of transformations. Warp transformation refinement requires extracting the reference PRNU pattern and the warped PRNU pattern from flat scenes (i.e., with uniform color content), that will be used to compare PCE. The more images that are used to compute the PRNUs, the more accurate will be the refinement.

Further details can be found in our paper (Bellavia, et al., 2019).

To have a deeper understanding of the transformations applied during EIS, we designed and built an ad-hoc structure made by a cubic metal frame, with a plywood panel on each face except one to strengthen the structure so as to decrease oscillations when moved or shaken (see Figure 10). On the empty face there is a thin grid made by a stretched nylon thread, whose intersection are evidenced by markers. On the opposite face to the grid, in the center, there is a hole on which the device acquisition sensor under test will be firmly held. When attacked to the structure, the device becomes integral with the grid. Each grid marker is virtually anchored to a location inside the device sensor matrix grid, so that markers seen through warped EIS frame can be mapped back into the sensor matrix grid. Knowing the marker correspondences with respect to the reference frame enables finding the EIS warping transformation for the current frame.

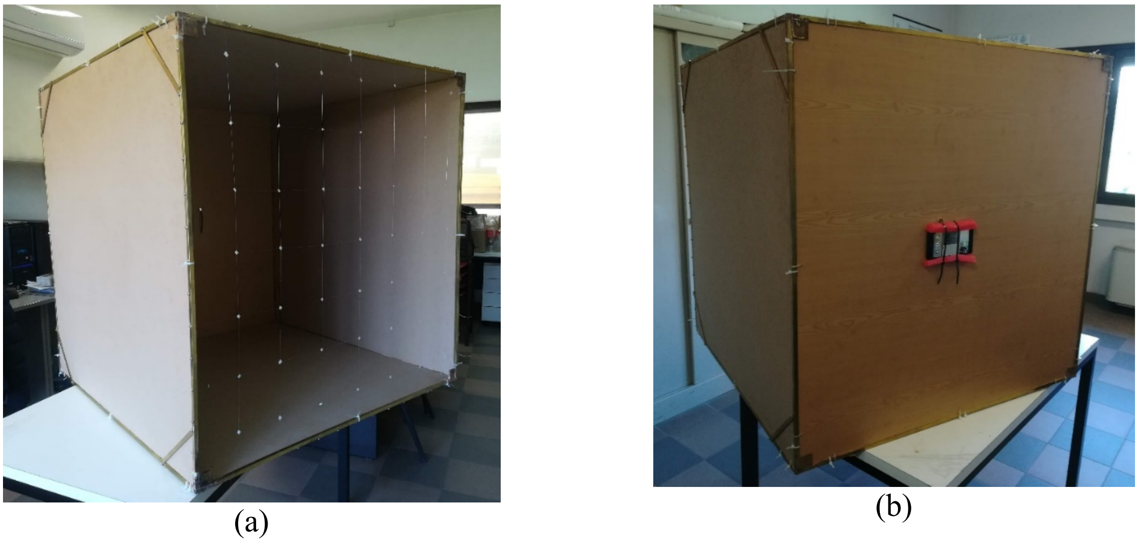


Figure 10: Front side and (a) back side (b) of the structure. Note the device placed in (best viewed in color and zoomed in)

To obtain a frame by frame correspondence between the markers, an automatic tracking system was developed. Due to the small size of the markers for avoiding interferences with EIS system, existing tracking solutions do not work. To solve this issue an ad-hoc tracking system was developed based on keypoint matching. To improve the matching accuracy, the putative corresponding keypoints are constrained to be inside a circular window of 50 pixel radius from the marker keypoints. The process is further refined using RANSAC to estimate planar homographies between grid corners of frame pairs.

4.2.3 PRNU re-synchronization using deep neural networks

PRNU matching for camera identification requires a perfect alignment between the probe noise residual and the device fingerprint (typically obtained from several flat-field images). Except for translation that can be recovered while computing PCE, rotation and scale transforms, applied to the probe image, need to be recovered before evaluating the correlation with the fingerprint, otherwise low PCE values are obtained: Indeed, even slight rotations or small scale changes are sufficient to spoil the camera identification task. To the best of our knowledge, the only method to solve this problem is to use brute force search, by warping the transformed probe with all possible values of angles and scales and computing PCE: the maximum PCE achieved identifies the transformation applied to the probe. This solution has two main drawbacks: (i) at first it is computationally expensive, having to evaluate a high number of correlations, to the point that it is practically unusable when dealing with huge datasets; (ii) secondly, there are high chances to get false positive or missing detections if few angle/scale pairs are tested.

A solution to this problem could have been provided by phase correlation of Fourier magnitude spectra in a log-polar representation (Reddy & Chatterji, 1996). However, after some testing, we discarded this option since results were not sufficiently reliable and computational times were still too high.

We resorted then to deep learning approaches. In (De Tone, et al., 2016) a CNN to compute homography transformations between natural images was presented. By moving from homography to similarity (considering scale and rotation only, without translation), we tried to use this method to register a device fingerprint with a residual noise extracted from a rotated and scaled probe.

The net was then trained on single devices from the VISION dataset (Shullani, et al., 2017). For each device, at first, we split the images in three subsets: training (70%), validation (10%) and test (20%). From the training subset we sampled 500,000 image patches, 20,000 from the validation subset (for more details on the patch sampling see (De Tone, et al., 2016)). The net was trained for 15 epochs, using a batch of 50 patches, repeated 10,000 times for each epoch. Validation patches were used to perform early stopping if the residual of the loss function rises with respect to the training.

4.3 Results

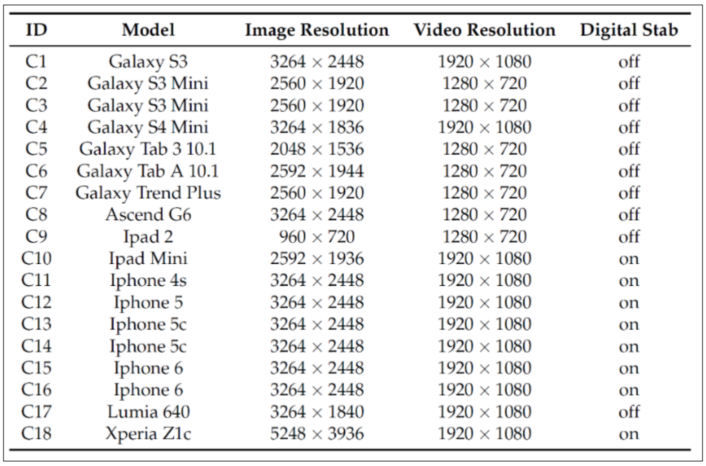
As done for the Methodology section above, here results are organized in different sub-sections, each one devoted to a specific PRNU related topic.

Note that on these topics we cannot provide results on the MediFor evaluation dataset, since the designed algorithms were not sufficiently developed at that time. On the other hand, we present several results and interesting insights on these particular PRNU related problems.

4.3.1 Results for hybrid reference based camera identification

We tested the proposed technique on a subpart of the VISION dataset (Shullani, et al., 2017), consisting of 1978 flatfield images, 3311 images of natural scenes and 339 videos captured by 18 devices from different brands (Apple, Samsung, Huawei, Microsoft, Sony). The dataset also provides the corresponding Facebook images, in both low quality (LQ) and high quality (HQ), and the corresponding YouTube videos. These contents have been used for testing the performance on the social media platforms. The tests have been carried out on a subpart of the VISION Dataset since the computational time needed to test all the devices was too high. We considered both smartphones and tablets depicting pictures and videos acquired with the default device settings that, for some models, include the automatic DV stabilization (see the VISION Dataset for details).

Figure 11 summarizes the considered models, their standard image and video resolution and whether the digital stabilization was active on the device. From now on we will refer to these devices with the names C1, . . ., C18 as defined in Figure 11. For each device we considered at least: On the reference side: 100 flat-field images depicting skies or walls; 150 images of indoor and outdoor scenes; 1 video



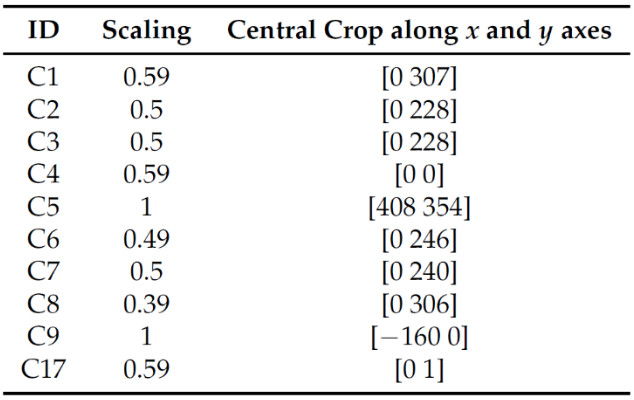
ID	Model	Image Resolution	Video Resolution	Digital Stab
C1	Galaxy S3	3264 × 2448	1920 × 1080	off
C2	Galaxy S3 Mini	2560 × 1920	1280 × 720	off
C3	Galaxy S3 Mini	2560 × 1920	1280 × 720	off
C4	Galaxy S4 Mini	3264 × 1836	1920 × 1080	off
C5	Galaxy Tab 3 10.1	2048 × 1536	1280 × 720	off
C6	Galaxy Tab A 10.1	2592 × 1944	1280 × 720	off
C7	Galaxy Trend Plus	2560 × 1920	1280 × 720	off
C8	Ascend G6	3264 × 2448	1280 × 720	off
C9	Ipad 2	960 × 720	1280 × 720	off
C10	Ipad Mini	2592 × 1936	1920 × 1080	on
C11	Iphone 4s	3264 × 2448	1920 × 1080	on
C12	Iphone 5	3264 × 2448	1920 × 1080	on
C13	Iphone 5c	3264 × 2448	1920 × 1080	on
C14	Iphone 5c	3264 × 2448	1920 × 1080	on
C15	Iphone 6	3264 × 2448	1920 × 1080	on
C16	Iphone 6	3264 × 2448	1920 × 1080	on
C17	Lumia 640	3264 × 1840	1920 × 1080	off
C18	Xperia Z1c	5248 × 3936	1920 × 1080	on

Figure 11: Considered devices with their default resolution settings for image and video acquisition

of the sky captured with slow camera movement, longer than 10 s. On the query side: videos of flat surfaces, indoor scenes and outdoor scenes. For each of the video categories (flat, indoor and outdoor) at least 3 different videos have been captured considering the three different scenarios available in the Dataset: (i) still camera, (ii) walking operator and (iii) panning and rotating camera. We will refer to them as still, move and panrot videos respectively. Thus, each device has at least 9 videos, each one lasting more than 60 s.

The experimental section consists of two main contributions: (i) we determine the cropping and scaling parameters applied by each device model in the considered set; (ii) we verify that, in the case of non-stabilized video, the performance of the hybrid approach is comparable with the source identification based on a video reference. Experiments were also performed on digitally stabilized videos and social media contents. Complete results are reported in (Iuliani, 2019).

The scaling and cropping factors applied by each device were derived by registering the reference video fingerprint. For each device, we estimated image fingerprints by means of 100 images randomly chosen from the flat-field pictures. For non-stabilized videos, video fingerprint was derived by means of the first 100 frames of the reference video available for that device. We opted for using all frames of the video instead of limiting to intra-coded frames (I-frames) only; this choice may slightly limit the performance in the non-stabilized video case, but it helps to greatly reduce the computational effort in the stabilized video case. Indeed, registration parameters of two consecutive frames in a stabilized video are expected to be closer than the ones of two distant I-frames; on the contrary, using only I-frames would force us to reboot the brute force search each time. In our implementation, once registration parameters are found on a frame, they are used to initialize the parameters in the next frame exploiting their proximity in time.



ID	Scaling	Central Crop along x and y axes
C1	0.59	[0 307]
C2	0.5	[0 228]
C3	0.5	[0 228]
C4	0.59	[0 0]
C5	1	[408 354]
C6	0.49	[0 246]
C7	0.5	[0 240]
C8	0.39	[0 306]
C9	1	[−160 0]
C17	0.59	[0 1]

Figure 12: Rescaling and cropping parameters linking image and video Sensor Pattern Noises (SPNs) for the considered devices, in absence of in-camera digital stabilization.

Figure 12 reports the estimated cropping parameters (in terms of coordinates of the upper-left corner of the cropped area along x and y axes, whereas the right down corner is derived by the video size) and the scaling factor, maximizing the PCE. For instance, C1 image fingerprint should be scaled by a factor of 0.59 and cropped on the upper left side of 307 pixels along the y axis to match the video fingerprint; C9 is a pretty unique case in which the full frame is applied for video and is left and right cropped by 160 pixels to capture images. In the case of stabilized videos, the cropping and scaling factors change in time, with possible rotation applied too. For these devices we thus determined the registration parameters of the first 10 frames of the available video reference; the main statistics are reported in Figure 13.

The information provided in Figure 13 can be exploited to reduce the parameter search space in case of source identification of digitally stabilized videos. Indeed, an exhaustive search of all possible scaling and rotation parameters, required in a blind analysis, would be intractable on a large scale: in our tests a totally blind search can take up to 10 min per frame on a modern average-powered computer, while the informed search reduces the time to less than a minute for stabilized videos and a few seconds for non-stabilized videos.

ID	Scaling	Central Crop along x and y	Rotation (CCW)
C10	[0.806 **0.815** 0.821]	[243 **256** 261] [86 **100** 103]	[−0.2 **0** 0.2]
C11	[0.748 **0.750** 0.753]	[380 **388** 392] [250 **258** 265]	[−0.2 **0** 0.2]
C12	[0.684 **0.689** 0.691]	[287 **294** 304] [135 **147** 165]	[−0.2 **0** 0.6]
C13	[0.681 **0.686** 0.691]	[301 **318** 327] [160 **181** 195]	[−0.4 **0** 1]
C14	[0.686 **0.686** 0.689]	[261 **301** 304] [119 **161** 165]	[−0.4 **0** 0]
C15	[0.696 **0.703** 0.713]	[298 **322** 345] [172 **190** 218]	[−0.2 **0.2** 1.6]
C16	[0.703 **0.706** 0.708]	[315 **323** 333] [178 **187** 201]	[−0.2 **0.2** 0.4]
C18	[0.381 **0.384** 0.387]	[548 **562** 574] [116 **121** 126]	[0 **0** 0]

Figure 13: Rescaling and cropping parameters that link image and video SPNs for the considered devices

Now we compare the proposed technique with the state of the art approach (Chen, et al., 2007), where the fingerprint is derived by estimating the SPN from a reference video. The comparison is only meaningful for non-stabilized devices. For each device, the reference fingerprints were derived respectively from the first 100 natural reference images (for the proposed method) and from the first 100 frames of the reference video (for the video reference approach). Given a video query, the fingerprint to be tested was derived by the first 100 frames and compared to the reference image and video fingerprints respectively using the cropping and scaling parameters expected for the candidate device. In Figure 14 we report for each device: (i) the statistics of matching pairs (blue and pink represent image and video reference respectively); (ii) the statistics for mismatching cases (in red).

The plot shows that distributions can be perfectly separated when the reference is estimated from images (100% accuracy), while in the video reference case the accuracy is 99.5%, confirming that performance is comparable.

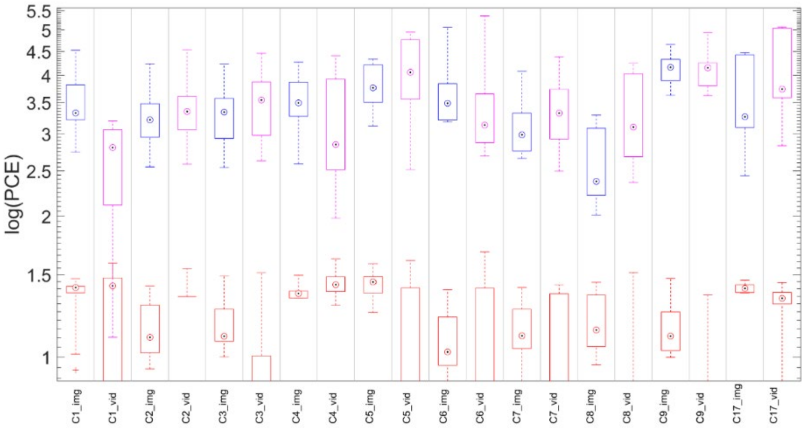


Figure 14: Matching and mismatching statistics. On each box, the central mark indicates the median, and the bottom and top edges of the box indicate the 25th and 75th percentiles respectively.

We also performed analysis on stabilized videos and social media contents. Technical details and results are reported in (Iuliani, 2019).

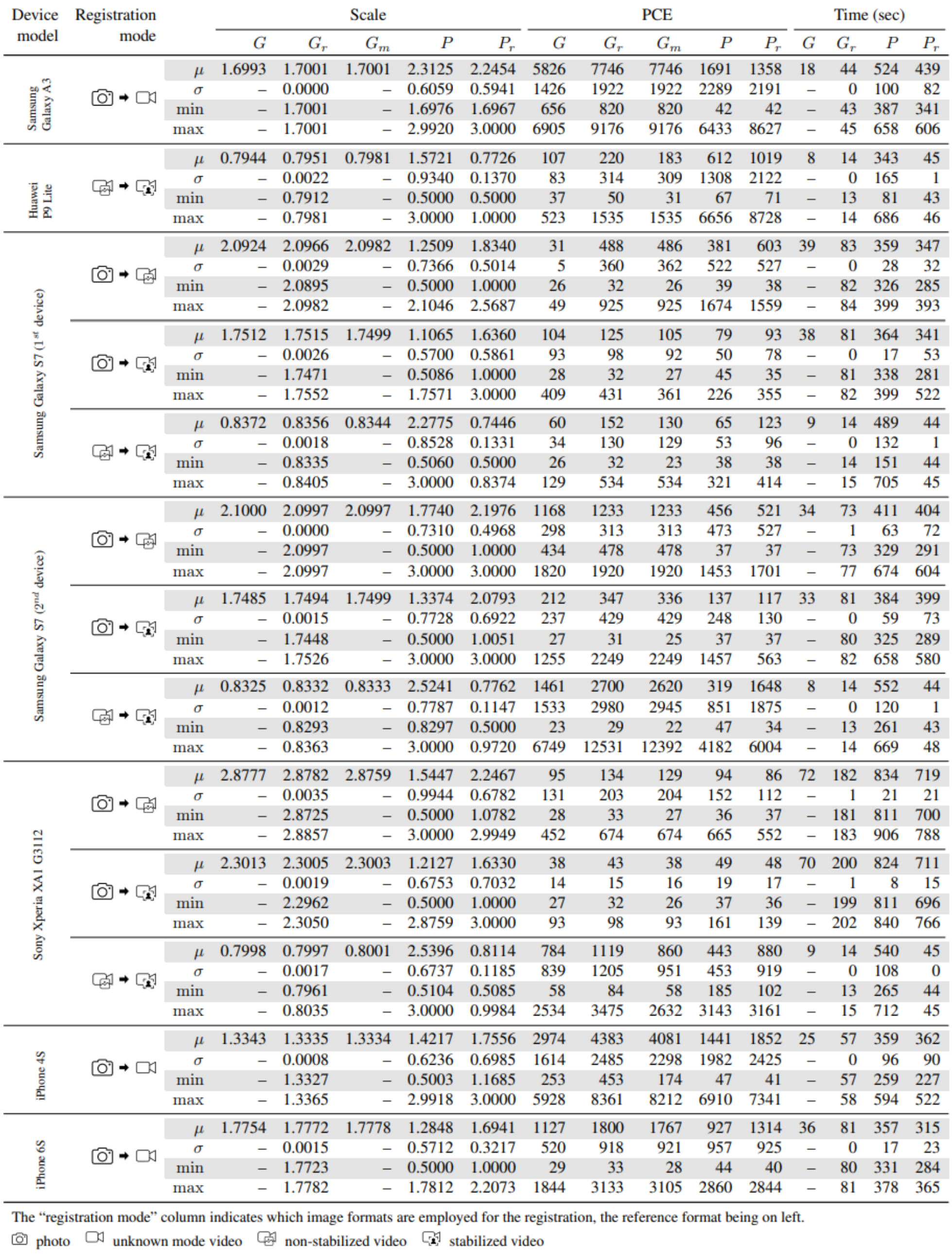
4.3.2 Results on understanding Electronic Image Stabilization

The proposed PRNU pattern registration approach is compared on seven different devices against particle swarm optimization (Mandelli, et al., 2019), which provides better accuracy and computational efficiency than brute-force approaches. For each device, the PRNU pattern of video I-frames from flat homogeneous scene content is warped according to the transformation parameters found by the corresponding method into the reference PRNU pattern. PCE between the warped and reference PRNU is then evaluated. The reference PRNU pattern is extracted from photos at native resolution but also from video I-frames acquired with EIS off when the source video to check was acquired with EIS on. Smooth video paths are assumed, so no rotations were considered. For each device, tested format and compared method, Table 7 reports the estimated scale, the accuracy in terms of PCE, and the running time. Results are presented in terms of mean μ, the standard deviation σ, and the minimum and maximum statistics. The proposed scene content PRNU alignment and its refinement are indicated as G and G_r, respectively. Additionally, G_m represents the results obtained by averaging G_r scales while discarding video I-frames with low PCE values (i.e., less than 50) on G, as a fast way to skip unreliable frames. For particle swarm, implemented using the Matlab built-in function, two different setups are evaluated. In detail, setup P uses 35 particles and a scale search range in [0.5,3], while setup P_r uses 30 particles and a scale search range in [1,3] and [0.5,1], respectively when the reference PRNU pattern is extracted from native full resolution photos or video frames captured with EIS off. Notice that the total running time for G_r is obtained by adding the corresponding columns G and G_r in the table.

The mean PCE value obtained with the scene content registration method G only is in most cases quite accurate, even without scale refinement (method G_r). The average registration G_m gives values very close to those given by G_r. The almost identical scale values obtained with the two different Samsung S7 devices witness that the warping transformation between the different image formats do not change among devices of the same model. This is quite reasonable since the warping process is not analog but digital, unlikely acquisition. This implies that G_m warping information can be used with other devices of the same model, avoiding acquiring each time ad-hoc static scene images or videos. Moreover, transformations across the different image acquisition formats can be concatenated without any accuracy degradation. Concerning particle swarm optimization, P_r results are usually more accurate and reliable than those obtained with P confirming that the particle swarm approach can lead to unstable or even wrong solutions if no clues about the allowable transformation parameter range are available.

Scene-based PRNU pattern registration is in general more accurate and reliable than that obtained by particle swarm optimization, also considering the lower excursion range in the scale and PCE values by inspecting the standard deviation, minimum and maximum related values. Notice also that in this case P obtains the highest average PCE value after P_r, but it is more distant in terms of the retrieved scale from P_r than scene content based methods, underlining some accidental inconsistencies that may happens due to the stochastic nature of PRNU. Analogous considerations about consistency and stability of the scales and PCE values hold for the Samsung Galaxy S7 (2^{nd} device, mapping from photos to I-frames with EIS off) and the Sony Xperia XA1 G3112 (mapping from photos to I-frames with EIS on), whose average PCE values can be slightly better for the P_r particle swarm than the scene content approaches.

Table 7: PRNU geometric registration results, compared against (Mandelli, et al., 2019)



Device model	Registration mode		Scale					PCE					Time (sec)			
			G	G_r	G_m	P	P_r	G	G_r	G_m	P	P_r	G	G_r	P	P_r
Samsung Galaxy A3	📷 → 🎥	μ	1.6993	1.7001	1.7001	2.3125	2.2454	5826	7746	7746	1691	1358	18	44	524	439
		σ	–	0.0000	–	0.6059	0.5941	1426	1922	1922	2289	2191	–	0	100	82
		min	–	1.7001	–	1.6976	1.6967	656	820	820	42	42	–	43	387	341
		max	–	1.7001	–	2.9920	3.0000	6905	9176	9176	6433	8627	–	45	658	606
Huawei P9 Lite	🎞 → 🎬	μ	0.7944	0.7951	0.7981	1.5721	0.7726	107	220	183	612	1019	8	14	343	45
		σ	–	0.0022	–	0.9340	0.1370	83	314	309	1308	2122	–	0	165	1
		min	–	0.7912	–	0.5000	0.5000	37	50	31	67	71	–	13	81	43
		max	–	0.7981	–	3.0000	1.0000	523	1535	1535	6656	8728	–	14	686	46
Samsung Galaxy S7 (1st device)	📷 → 🎞	μ	2.0924	2.0966	2.0982	1.2509	1.8340	31	488	486	381	603	39	83	359	347
		σ	–	0.0029	–	0.7366	0.5014	5	360	362	522	527	–	0	28	32
		min	–	2.0895	–	0.5000	1.0000	26	32	26	39	38	–	82	326	285
		max	–	2.0982	–	2.1046	2.5687	49	925	925	1674	1559	–	84	399	393
	📷 → 🎬	μ	1.7512	1.7515	1.7499	1.1065	1.6360	104	125	105	79	93	38	81	364	341
		σ	–	0.0026	–	0.5700	0.5861	93	98	92	50	78	–	0	17	53
		min	–	1.7471	–	0.5086	1.0000	28	32	27	45	35	–	81	338	281
		max	–	1.7552	–	1.7571	3.0000	409	431	361	226	355	–	82	399	522
	🎞 → 🎬	μ	0.8372	0.8356	0.8344	2.2775	0.7446	60	152	130	65	123	9	14	489	44
		σ	–	0.0018	–	0.8528	0.1331	34	130	129	53	96	–	0	132	1
		min	–	0.8335	–	0.5060	0.5000	26	32	23	38	38	–	14	151	44
		max	–	0.8405	–	3.0000	0.8374	129	534	534	321	414	–	15	705	45
Samsung Galaxy S7 (2nd device)	📷 → 🎞	μ	2.1000	2.0997	2.0997	1.7740	2.1976	1168	1233	1233	456	521	34	73	411	404
		σ	–	0.0000	–	0.7310	0.4968	298	313	313	473	527	–	1	63	72
		min	–	2.0997	–	0.5000	1.0000	434	478	478	37	37	–	73	329	291
		max	–	2.0997	–	3.0000	3.0000	1820	1920	1920	1453	1701	–	77	674	604
	📷 → 🎬	μ	1.7485	1.7494	1.7499	1.3374	2.0793	212	347	336	137	117	33	81	384	399
		σ	–	0.0015	–	0.7728	0.6922	237	429	429	248	130	–	0	59	73
		min	–	1.7448	–	0.5000	1.0051	27	31	25	37	37	–	80	325	289
		max	–	1.7526	–	3.0000	3.0000	1255	2249	2249	1457	563	–	82	658	580
	🎞 → 🎬	μ	0.8325	0.8332	0.8333	2.5241	0.7762	1461	2700	2620	319	1648	8	14	552	44
		σ	–	0.0012	–	0.7787	0.1147	1533	2980	2945	851	1875	–	0	120	1
		min	–	0.8293	–	0.8297	0.5000	23	29	22	47	34	–	13	261	43
		max	–	0.8363	–	3.0000	0.9720	6749	12531	12392	4182	6004	–	14	669	48
Sony Xperia XA1 G3112	📷 → 🎞	μ	2.8777	2.8782	2.8759	1.5447	2.2467	95	134	129	94	86	72	182	834	719
		σ	–	0.0035	–	0.9944	0.6782	131	203	204	152	112	–	1	21	21
		min	–	2.8725	–	0.5000	1.0782	28	33	27	36	37	–	181	811	700
		max	–	2.8857	–	3.0000	2.9949	452	674	674	665	552	–	183	906	788
	📷 → 🎬	μ	2.3013	2.3005	2.3003	1.2127	1.6330	38	43	38	49	48	70	200	824	711
		σ	–	0.0019	–	0.6753	0.7032	14	15	16	19	17	–	1	8	15
		min	–	2.2962	–	0.5000	1.0000	27	32	26	37	36	–	199	811	696
		max	–	2.3050	–	2.8759	3.0000	93	98	93	161	139	–	202	840	766
	🎞 → 🎬	μ	0.7998	0.7997	0.8001	2.5396	0.8114	784	1119	860	443	880	9	14	540	45
		σ	–	0.0017	–	0.6737	0.1185	839	1205	951	453	919	–	0	108	0
		min	–	0.7961	–	0.5104	0.5085	58	84	58	185	102	–	13	265	44
		max	–	0.8035	–	3.0000	0.9984	2534	3475	2632	3143	3161	–	15	712	45
iPhone 4S	📷 → 🎥	μ	1.3343	1.3335	1.3334	1.4217	1.7556	2974	4383	4081	1441	1852	25	57	359	362
		σ	–	0.0008	–	0.6236	0.6985	1614	2485	2298	1982	2425	–	0	96	90
		min	–	1.3327	–	0.5003	1.1685	253	453	174	47	41	–	57	259	227
		max	–	1.3365	–	2.9918	3.0000	5928	8361	8212	6910	7341	–	58	594	522
iPhone 6S	📷 → 🎥	μ	1.7754	1.7772	1.7778	1.2848	1.6941	1127	1800	1767	927	1314	36	81	357	315
		σ	–	0.0015	–	0.5712	0.3217	520	918	921	957	925	–	0	17	23
		min	–	1.7723	–	0.5000	1.0000	29	33	28	44	40	–	80	331	284
		max	–	1.7782	–	1.7812	2.2073	1844	3133	3105	2860	2844	–	81	378	365

The "registration mode" column indicates which image formats are employed for the registration, the reference format being on left.

📷 photo 🎥 unknown mode video 🎞 non-stabilized video 🎬 stabilized video

Concerning running times, scene-based registration G is very fast and even by summing up the further refinement step G_r, the approach is faster than particle swarm optimization. In particular, our full approach G_r is about four times faster than particle swarm optimization, except for non-stabilized to stabilized video PRNU registration with setup P_r, for which our approach is only twice faster. Note that running times depend on image resolution and hence on the scale search range. Clearly, particle swarm accuracy can be improved by using more particles in the setup, yet computation time would increase accordingly.

Regarding the study made using our developed structure (see Figure 10), two mid-range smartphones, the Huawei P9 Lite and the Xiaomi M2 A1, were considered for analyzing EIS. The testing video sequences were obtained by moving and shaking the devices installed on the structure in front of a fixed background, and additionally introducing moving foreground objects of different sizes (i.e. walking or jumping people and fluttering flyers). For each frame, the homography obtained by tracking the grid markers is employed to revert back the frame EIS transformation.

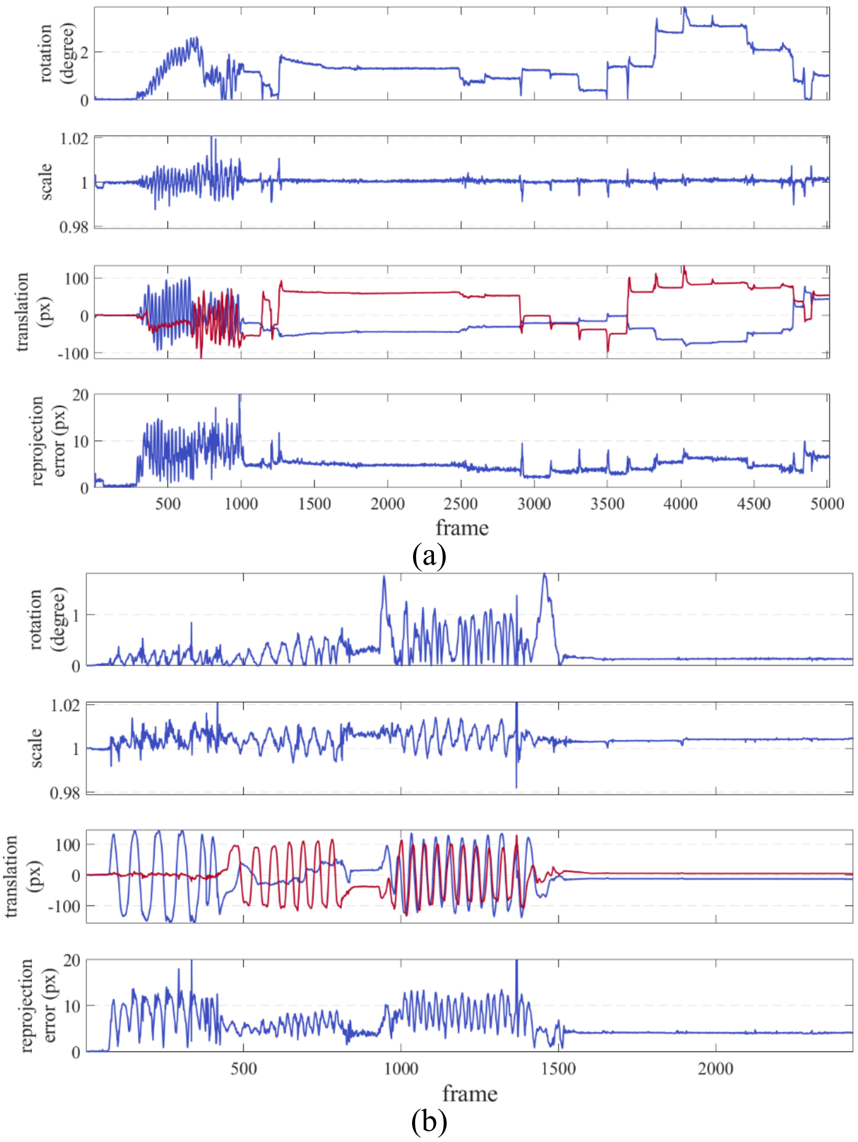


Figure 15: Decomposition of EIS frame transformation according to a similarity for (a) Huawei P9 Lite and (b) Xiaomi M2 A1 videos (best viewed in color and zoomed in).

The tracking is quite stable, except in some cases due to motion blur effects that decrease keypoint localization accuracy and for some unabsorbed, non-rigid oscillation of the structure with respect to the camera. Figure 15 depicts for each analyzed sequence the decomposition of the EIS frame transformation into a similarity. In addition to each component of the similarity transform, it also indicated the reprojection error. This reprojection error is fully compatible with the keypoint localization accuracy of the scene content of the current frame. Moreover, the minimal variation of the scale component, that is associated to the temporally decrement in the keypoint accuracy discussed above, suggests that frame transformations inside an EIS video are only metric, i.e. only rotation and translation are involved. In addition, no rotation of more than 5 degrees was observed.

Both scale constraint and rotation limits can be exploited for designing new PRNU registration methods.

Furthermore, from the analysis carried out, it emerges that different device models use distinct EIS implementations. In particular it comes out that Huawei P9 Lite triggers EIS according to scene visual flow, opposing to the Xiaomi M2 A1, for which EIS is based on physical movement sensors (i.e., gyroscopes or accelerometers). Moreover, unlikely the Xiaomi M2 A1, it can be observed that EIS frame rotation steps seem quantized for the Huawei P9 Lite, maybe due to the usage internally of Look-up Tables (LUT) for an efficient computation of the frame warp. Finally, the Huawei P9 Lite tends to maintain the last frame transformation over the next frames even if the condition that has triggered EIS disappears, while the Xiaomi M1 A2 in this case tends to smoothly come back to the original reference status. The last seems to be the common behavior of most devices and explains why PRNU alignment can often work, regardless of EIS, on sufficient long sequences, as most of the frames will result in being aligned to the reference first frame.

4.3.3 Results for PRNU re-synchronization using deep neural networks

Once training was completed, we tested each net on 100,000 samples extracted from the test subset. Note that, in this test we evaluate the net performance considering only probes coming from the same device (i.e. we consider only the case of correct detection or missing detection). Hereafter Table 8 shows the percentage of correctly registered probes – i.e. probes that, after registration, obtain PCE scores higher than 60 or 100 – for 11 devices of the VISION dataset.

Table 8: results on sigle device.

Device	PCE > 60 (%)	PCE
D01_Samsung_GalaxyS3Mini	87.6	84.3
D02_Apple_iPhone4s	60.2	51.9
D04_LG_D290	71.0	64.1
D05_Apple_iPhone5c	90.9	88.9
D07_Lenovo_P70A	91.4	88.6
D08_Samsung_GalaxyTab3	82.4	79.0
D09_Apple_iPhone4	93.6	92.2
D11_Samsung_GalaxyS3	83.6	80.1
D12_Sony_XperiaZ1Compact	86.0	83.5
D13_Apple_iPad2	89.7	87.8
D15_Apple_iPhone6	88.5	85.1

As can be seen, for most of the devices under test the achieved performance was promising. We then tried to perform an ALL-vs-ALL test, in which a given rotated probe from a device is tested against all the devices. Hereafter, in Figure 16 (left) we report the obtained confusion matrix. The results show that, except for D02 and D04 (that obtained lower performance even in the single device test), most of the devices obtained high percentage of correct detection, few false positives, and some false negatives. To improve the results, we tried to perform a multiple patch sampling: given a transformed probe, we extract three different patches to give to the net, obtaining in output three possible transformations: each transformation is tested using PCE, and the best score achieved is retained. Hereafter the relative confusion matrix Figure 16 (right). As shown, in this way we obtained almost zero false positive (only 5) and improved detection rates, at the expense of increased computational times (about 3 times slower due to the multiple computation of PCE).

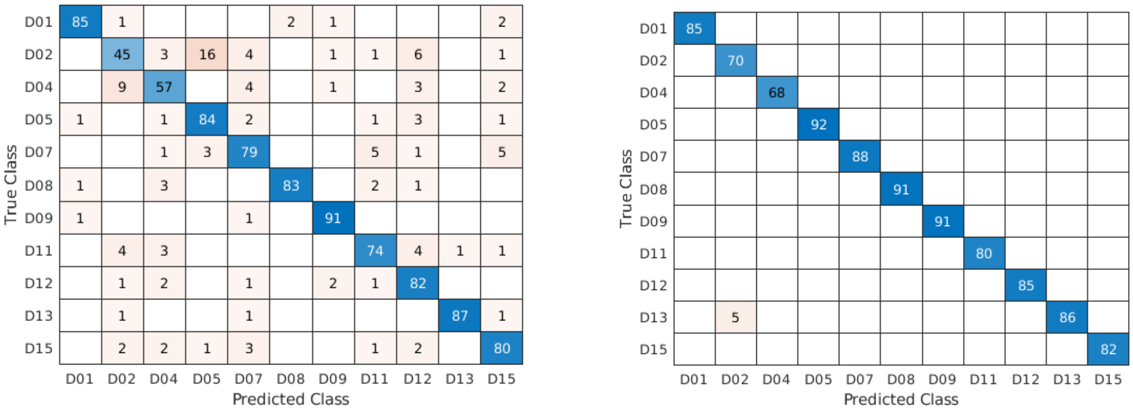


Figure 16: ALL-vs-ALL confusion matrix, with single patch sampling (left) and using learned registration and multi-patch sampling (right) .

4.3.4 Conclusions

We proposed a hybrid approach to video source identification using a reference fingerprint derived from still images. We showed that, in the case of non-stabilized videos, the hybrid approach yields performance comparable with or even better than the current state-of-the-art strategy, which uses a video to compute the reference pattern. Our approach allows reliable source identification even for videos produced by devices that enforce digital in-camera stabilization (e.g., all recent Apple devices), for which a non-stabilized reference is not available. The main limitation of the proposed approach is the need for a brute force search for determining scale (and, in the case of stabilized devices, rotation) when no information on the tested device is available.

Then, we presented some novel idea to work with PRNU pattern alignment. Firstly, we proposed a technique to estimate the scale changes among different capture modalities (image, video and stabilized video) that uses the image scene content to obtain an initial registration using local image descriptors and can be further refined by maximizing the PRNU correlation. This solution has shown to be more reliable, more accurate and faster than existing approaches based on brute-force and particle swarm optimization. Then, we presented a solution to revert-back the transformation introduced by a device during video stabilization. From experimental evidence we argued that stabilization uses only scale, rotation, and translation, without more complex transformation.

Finally, we presented some initial results on PRNU re-synchronization using deep learning. Inspired by works on homography estimation with convolutional neural networks on natural images, we tried to apply this approach on PRNU signals, limiting the estimation to scale and rotation. After training the net on single devices, we were able to recover the applied transformation in most of the cases, for most of the tested devices. Also, we tested the performance of this solution in an *all-vs-all* test, were a rotated and scaled probe is compared with all the devices. For each of the reference devices, at first the probe is given as input to the respective net, than the estimated transform is applied to the probe and finally the PCE w.r.t. the device fingerprint is computed. As can be seen from the reported results, the approach seems very promising, in particular if we sample multiple patches from the probe, reducing false positive and improving the correct detection rates.

5 References

10918-1, I. E. C., 1992. Information technology. Digital compression and coding of continuous-tone still images. *ITU-T Recommendation T.81.*

Aitken, C. C. G. et al., 2015. ENFSI Guideline for Evaluative Reporting in Forensic Science. *European Network of Forensic Science Institutes (ENFSI).*

Anon., s.d. *Clip Grab.* [Online] Available at: \url{https://clipgrab.org/}

Anon., s.d. Exiftool. *http://www.sno.phy.queensu.ca/ phil/exiftool.*

Anon., s.d. Express.js. *http://www.http://expressjs.com.*

Anon., s.d. Ffmpeg. *https://ffmpeg.org.*

Anon., s.d. *Keepvid.* [Online] Available at: www.keepvid.com

Anon., s.d. Node.js. *https://nodejs.org.*

Anon., s.d. *Statcounter: GlobalStats 1999-2020.* s.l.:s.n.

Apache, s.d. Java MP4 Parser. *http://www.github.com/sannies/mp4parser.*

Apache, s.d. JDOM. *http://www.jdom.com.*

Apple Computer, Inc., 2001. Quicktime file format.

Bakas, J. & Naskar, R., 2018. *A Digital Forensic Technique for Inter–Frame Video Forgery Detection Based on 3D CNN.* s.l., s.n., p. 304–317.

Basri, R. & Jacobs, D. W., 2003. Lambertian reflectance and linear subspaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 2, Volume 25, pp. 218-233.

Bayram, S., Sencar, H. T., Memon, N. & Avcibas, I., 2005. *Source camera identification based on CFA interpolation.* s.l., s.n., p. III–69.

Beck, M., 2015. *Reversal of Facebook: Photo posts now drive lowest organic reach.* s.l.:s.n.

Bellavia, F. & Colombo, C., 2018. Rethinking the sGLOH Descriptor. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Volume 40, p. 931–944.

Bellavia, F., Tegolo, D. & Valenti, C., 2011. Improving Harris corner selection strategy. *IET Computer Vision,* Volume 5, p. 86–96.

Bertini, F. et al., 2016. *Social media investigations using shared photos.* s.l., s.n., p. 47.

Breiman, L., 2017. *Classification and regression trees.* s.l.:Routledge.

Bruna, A. R., Messina, G. & Battiato, S., 2011. *Crop detection through blocking artefacts analysis.* s.l., s.n., p. 650–659.

Cao, C. et al., 2014. FaceWarehouse: A 3D Facial Expression Database for Visual Computing. *IEEE Transactions on Visualization and Computer Graphics,* 3, Volume 20, p. 413–425.

Castiglione, A., Cattaneo, G., Cembalo, M. & Petrillo, U. F., 2013. Experimentations with source camera identification and online social networks. *Journal of Ambient Intelligence and Humanized Computing,* Volume 4, p. 265–274.

Cattaneo, G., Roscigno, G. & Petrillo, U. F., 2014. *A scalable approach to source camera identification over Hadoop.* s.l., s.n., p. 366–373.

Chen, M., Fridrich, J., Goljan, M. & Lukáš, J., 2007. *Source digital camcorder identification using sensor photo response non-uniformity.* s.l., s.n., p. 65051G.

Chen, M., Fridrich, J., Goljan, M. & Lukas, J., 2008. Determining Image Origin and Integrity Using Sensor Noise. *IEEE Transactions on Information Forensics and Security,* Volume 3, pp. 74-90.

Chen, M., Fridrich, J., Goljan, M. & Lukáš, J., 2008. Determining image origin and integrity using sensor noise. *IEEE Transactions on Information Forensics and Security,* Volume 3, p. 74–90.

Chen, S., Pande, A., Zeng, K. & Mohapatra, P., 2015. Live video forensics: source identification in lossy wireless networks. *IEEE Transactions on Information Forensics and Security,* Volume 10, p. 28–39.

Chuang, W.-H., Su, H. & Wu, M., 2011. *Exploring compression effects for improved source camera identification using strongly compressed video.* s.l., s.n., p. 1953–1956.

Cisco Visual Networking Index, 2016. Forecast and methodology, 2016-2021, white paper. *San Jose, CA, USA.*

Colombo, C., Comanducci, D. & Del Bimbo, A., 2006. *Camera Calibration with Two Arbitrary Coaxial Circles.* Berlin, Springer Berlin Heidelberg, p. 265–276.

Coughlan, J. M. & Yuille, A. L., 1999. *Manhattan World: compass direction from a single image by Bayesian inference.* s.l., s.n., pp. 941-947 vol.2.

D'Avino, D., Cozzolino, D., Poggi, G. & Verdoliva, L., 2017. Autoencoder with recurrent neural networks for video forgery detection. *Electronic Imaging,* Volume 2017, p. 92–99.

De Tone, D., Malisiewicz, T. & Rabinovich, A., 2016. Deep image homograpy estimation. *arXiv preprint.*

Deutscher, J., Isard, M. & MacCormick, J., 2002. *Automatic Camera Calibration from a Single Manhattan Image.* London, Springer-Verlag, p. 175–205.

Developer, A. I., 2017. *Working with HEIF and HEVC.* s.l.:s.n.

Ding, X. et al., 2017. Identification of Motion-Compensated Frame Rate Up-Conversion Based on Residual Signal. *IEEE Transactions on Circuits and Systems for Video Technology.*

Dirik, A. E., Sencar, H. T. & Memon, N., 2008. Digital single lens reflex camera identification from traces of sensor dust. *IEEE Transactions on Information Forensics and Security,* Volume 3, p. 539–552.

Drygajlo, A., 2007. Forensic automatic speaker recognition [Exploratory DSP]. *IEEE Signal Processing Magazine,* Volume 24, p. 132–135.

ENFSI MP2013 T6, 2013. *The development of a statistical software package for likelihood ratio calculations.* s.l.:s.n.

Fanfani, M. et al., 2019. FISH: Face Intensity-Shape Histogram representation for automatic face splicing detection. *Journal of Visual Communication and Image Representation,* 63(102586).

Fanfani, M. et al., 2020. A vision-based fully automated approach to robust image cropping detection. *Signal Processing: Image Communication,* 80(115629).

Farid, H., 2008. Digital image ballistics from JPEG quantization: a followup study. *[Technical Report TR2008–638] Department of Computer Science, Dartmouth College, Hanover, NH, USA.*

Felsenstein, J., 1981. Evolutionary trees from DNA sequences: a maximum likelihood approach. *Journal of molecular evolution,* Volume 17, p. 368–376.

Fischler, M. & Bolles, R., 1981. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM,* Volume 24, pp. 381-395.

Geradts, Z. J. et al., 2001. *Methods for identification of images acquired with digital cameras.* s.l., s.n., p. 505–512.

Giammarrusco, Z. P., 2014. *Source identification of high definition videos: a forensics analysis of downloaders and Youtube video compression using a group of action cameras.,* s.l.: s.n.

Gironi, A. et al., 2014. *A video forensic technique for detecting frame deletion and insertion.* s.l., s.n., p. 6226–6230.

Gloe, T., 2012. *Forensic analysis of ordered data structures on the example of JPEG files.* s.l., s.n., p. 139–144.

Gloe, T. & Böhme, R., 2010. The Dresden image database for benchmarking digital image forensics. *Journal of Digital Forensic Practice,* Volume 3, p. 150–159.

Gloe, T., Fischer, A. & Kirchner, M., 2014. Forensic analysis of video file formats. *Digital Investigation,* Volume 11, Supplement 1, pp. S68 - S76.

Goljan, M., 2009. Digital camera identification from images - Estimating false acceptance probability. In: *Digital watermarking.* s.l.:Springer, p. 454–468.

Goljan, M., Chen, M. & Fridrich, J. J., 2007. *Identifying common source digital camera from image pairs.* s.l., s.n., p. 125–128.

Goljan, M. & Fridrich, J., 2008. *Camera identification from cropped and scaled images.* s.l., s.n., p. 68190E.

Goljan, M., Fridrich, J. & Filler, T., 2009. *Large scale test of sensor fingerprint camera identification.* s.l., s.n., p. 72540I.

Goljan, M. & Fridrich, J. J., 2008. *Camera identification from cropped and scaled images.* s.l., s.n.

Güera, D. et al., 2019. *We Need No Pixels: Video Manipulation Detection Using Stream Descriptors.* s.l., s.n.

Guillou, E., Meneveaux, D., Maisel, E. & Bouatouch, K., 2000. Using vanishing points for camera calibration and coarse 3D reconstruction from a single image. *The Visual Computer,* Volume 16, p. 396–410.

Hoglund, T., Brolund, P. & Norell, K., 2011. *Identifying camcorders using noise patterns from video clips recorded with image stabilisation.* s.l., s.n., pp. 668-671.

Höglund, T., Brolund, P. & Norell, K., 2011. *Identifying camcorders using noise patterns from video clips recorded with image stabilisation.* s.l., s.n., p. 668–671.

Holt, C. R., 1987. Two-channel likelihood detectors for arbitrary linear channel distortion. *IEEE Transactions on Acoustics, Speech and Signal Processing,* Volume 35, p. 267–273.

ISO/IEC 14496, 2003. Information technology. coding of audio-visual objects, Part 14: MP4 file format.

ISO/IEC 14496, 2008. Information technology. Coding of audio-visual objects, Part 12: ISO base media file format, 3rd ed..

Iuliani, M., Fanfani, M., Colombo, C. & Piva, A., 2017. Reliability Assessment of Principal Point Estimates for Forensic Applications. *Journal of Visual Communication and Image Representation,* Volume 42.

Iuliani, M. F. M. S. D. &. P. A., 2019. Hybrid reference-based video source identification. *Sensors,* 3(19), p. 649.

Iuliani, M. et al., 2018. A video forensic framework for the unsupervised analysis of MP4-like file container. *IEEE Transactions on Information Forensics and Security,* Volume 14, p. 635–645.

JEITA CP-3451, 2002. Exchangeable image file format for digital still cameras: Exif version 2.2.

Johnson, M. K. & Farid, H., 2007. Exposing Digital Forgeries in Complex Lighting Environments. *Information Forensics and Security, IEEE Transactions on,* Volume 2, pp. 450-461.

Kee, E., Johnson, M. K. & Farid, H., 2011. Digital Image Authentication From JPEG Headers. *IEEE Transactions on Information Forensics and Security,* 9, Volume 6, pp. 1066-1075.

Kee, E., Johnson, M. K. & Farid, H., 2011. Digital Image Authentication From JPEG Headers. *IEEE Transactions on Information Forensics and Security,* 9, Volume 6, pp. 1066-1075.

Korus, P., 2017. Digital image integrity–a survey of protection and verification techniques. *Digital Signal Processing,* Volume 71, p. 1–26.

Liu, B.-b., Wei, X. & Yan, J., 2015. *Enhancing sensor pattern noise for source camera identification: an empirical evaluation.* s.l., ACM, p. 85–90.

Li, W., Yuan, Y. & Yu, N., 2009. Passive detection of doctored JPEG image via block artifact grid extraction. *Signal Processing,* Volume 89, p. 1821–1829.

Lukas, J., Fridrich, J. & Goljan, M., 2006. Digital camera identification from sensor pattern noise. *IEEE Transactions on Information Forensics and Security,* Volume 1, p. 205–214.

Mandelli, S., Bestagini, P., Verdoliva, L. & Tubaro, S., 2018. Facing device attribution problem for stabilized video sequences. *arXiv preprint arXiv:1811.01820.*

Mandelli, S., Bestagini, P., Verdoliva, L. & Tubaro, S., 2019. Facing Device Attribution Problem for Stabilized Video Sequences. *IEEE Transactions on Information Forensics and Security,* Volume 15, p. 14–27.

Marra, F., Poggi, G., Sansone, C. & Verdoliva, L., 2017. Blind PRNU-based image clustering for source identification. *IEEE Transactions on Information Forensics and Security,* Volume 12, p. 2197–2211.

Mathias, M., Benenson, R., Pedersoli, M. & Van Gool, L., 2014. *Face Detection without Bells and Whistles.* Cham, Springer International Publishing, p. 720–735.

Maxwell, R., 2016. *Camera vs. Smartphone: infographic shares the impact our smartphones have had on regular cameras.* s.l.:s.n.

Meuwly, D., Ramos, D. & Haraksim, R., 2017. A guideline for the validation of likelihood ratio methods used for forensic evidence evaluation. *Forensic Science International,* Volume 276, pp. 142-153.

Meuwly, D., Ramos, D. & Haraksim, R., 2017. A guideline for the validation of likelihood ratio methods used for forensic evidence evaluation. *Forensic Science International,* Volume 276, pp. 142-153.

Microsoft Developer Network, s.d. AVI RIFF file reference. *http://msdn.microsoft.com/en-us/library/ms779636(VS.85).aspx.*

Mihcak, M. K., Kozintsev, I. & Ramchandran, K., 1999. *Spatially adaptive statistical modeling of wavelet image coefficients and its application to denoising.* s.l., s.n., p. 3253–3256.

Milani, S. et al., 2012. An overview on video forensics. *APSIPA Transactions on Signal and Information Processing,* Volume 1.

Moltisanti, M., Paratore, A., Battiato, S. & Saravo, L., 2015. *Image manipulation on Facebook for forensics evidence.* s.l., s.n., p. 506–517.

Mondaini, N. et al., 2007. *Detection of malevolent changes in digital video for forensic applications.* s.l., s.n., p. 65050T.

Nordgaard, A. & Höglund, T., 2011. Assessment of approximate likelihood ratios from continuous distributions: a case study of digital camera identification. *Journal of forensic sciences,* Volume 56, p. 390–402.

Paysan, P. et al., 2009. *A 3D Face Model for Pose and Illumination Invariant Face Recognition.* s.l., s.n., pp. 296-301.

Pedregosa, F. et al., 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research,* Volume 12, p. 2825–2830.

Peng, B., Wang, W., Dong, J. & Tan, T., 2015. *Improved 3D lighting environment estimation for image forgery detection.* s.l., s.n., pp. 1-6.

Peng, B., Wang, W., Dong, J. & Tan, T., 2016. *Automatic detection of 3D lighting inconsistencies via a facial landmark based morphable model.* s.l., s.n., pp. 3932-3936.

Peng, B., Wang, W., Dong, J. & Tan, T., 2017. Optimized 3D Lighting Environment Estimation for Image Forgery Detection. *IEEE Transactions on Information Forensics and Security,* 2, Volume 12, pp. 479-494.

Peterson, T., 2016. *Facebook users are posting 75% more videos than last year.* s.l.:s.n.

Pflugfelder, R. & Bischof, H., 2005. *Online Auto-Calibration in Man-Made Worlds.* s.l., s.n., pp. 75-75.

Piva, A., 2013. An overview on image forensics. *ISRN Signal Processing.*

Piva, A., 2013. An Overview on Image Forensics. *ISRN Signal Processing,* Volume 2013, p. 1–22.

Posada, D. & Crandall, K. A., 1998. Modeltest: testing the model of DNA substitution.. *Bioinformatics (Oxford, England),* Volume 14, p. 817–818.

Qadir, G., Yahaya, S. & Ho, A. T. S., 2012. Surrey University library for forensic analysis (SULFA) of video content.

Quinlan, J. R., 1986. Induction of decision trees. *Machine learning,* Volume 1, p. 81–106.

Ramamoorthi, R. & Hanrahan, P., 2001. On the relationship between radiance and irradiance: determining the illumination from images of a convex Lambertian object. *J. Opt. Soc. Am. A,* 10, Volume 18, p. 2448–2459.

Reddy, B. S. & Chatterji, B. N., 1996. An FFT-based technique for translation, rotation, and scale-invariant image registration. *IEEE Transactions on Image Processing,* 5(8), pp. 1266-1271.

Reynolds, D. A., Quatieri, T. F. & Dunn, R. B., 2000. Speaker verification using adapted Gaussian mixture models. *Digital signal processing,* Volume 10, p. 19–41.

Rosa, A. D., Piva, A., Fontani, M. & Iuliani, M., 2014. Investigating multimedia contents. *2014 International Carnahan Conference on Security Technology (ICCST),* 10.pp. 1-6.

Row, D. & Reid, T. J., 2012. *Geometry, Perspective Drawing, and Mechanisms.* s.l.:World Scientific.

Safavian, S. R. & Landgrebe, D., 1991. A survey of decision tree classifier methodology. *IEEE transactions on systems, man, and cybernetics,* Volume 21, p. 660–674.

Scheelen, J. v. d. L., Geradts, Z. & Worring, M., 2012. Camera identification on Youtube. *Chinese Journal of Forensic Science,* Volume 5, p. 19–30.

Schütze, H., Manning, C. D. & Raghavan, P., 2008. *Introduction to information retrieval.* s.l., s.n., p. 260.

Shanableh, T., 2013. Detection of frame deletion for digital video forensics. *Digital Investigation,* Volume 10, pp. 350-360.

Shullani, D. et al., 2017. VISION: a video and image dataset for source identification. *EURASIP Journal on Information Security,* Volume 2017, p. 15.

Shullani, D. et al., 2017. VISION: a video and image dataset for source identification. *EURASIP Journal on Information Security,* Volume 2017, p. 15.

Stamm, M. C. & Liu, K. R., 2011. Anti-forensics of digital image compression. *IEEE Transactions on Information Forensics and Security,* Volume 6, p. 1050–1065.

Steinebach, M., Liu, H., Fan, P. & Katzenbeisser, S., 2010. *Cell phone camera ballistics: attacks and countermeasures.* s.l., s.n., p. 75420B.

Su, L., Li, C., Lai, Y. & Yang, J., 2017. A Fast Forgery Detection Algorithm based on Exponential-Fourier Moments for Video Region Duplication. *IEEE Transactions on Multimedia.*

SWGDE-SWGIT, 2017. *Best Practices for Image Content Authentication, Version: 1.0.* s.l.:s.n.

SWGDE-SWGIT, 2017. *SWGDE Best Practices for Maintaining the Integrity of Imagery, Version: 1.0.* s.l.:s.n.

Szeliski, R., 2010. *Computer Vision: Algorithms and Applications.* 1st a cura di New York, NY, USA: Springer-Verlag New York, Inc..

Taspinar, S., Mohanty, M. & Memon, N., 2016. *Source camera attribution using stabilized video.* s.l., s.n., p. 1–6.

Taspinar, S., Mohanty, M. & Memon, N., 2016. *Source camera attribution using stabilized video.* s.l., s.n., pp. 1-6.

Trigeorgis, G., Snape, P., Kokkinos, I. & Zafeiriou, S., 2017. *Face Normals In-the-Wild Using Fully Convolutional Networks.* s.l., s.n., pp. 340-349.

Valsesia, D., Coluccia, G., Bianchi, T. & Magli, E., 2015. Compressed fingerprint matching and camera Iientification via random projections. *IEEE Transactions on Information Forensics and Security,* 7, Volume 10, pp. 1472-1485.

Van Houten, W. & Geradts, Z., 2009. Source video camera identification for multiply compressed videos originating from Youtube. *Digital Investigation,* Volume 6, p. 48–60.

Vázquez-Padın, D. et al., 2012. *Detection of video double encoding with GOP size estimation.* s.l., s.n., p. 151–156.

Vázquez-Padın, D. et al., 2012. *Detection of video double encoding with GOP size estimation.* s.l., s.n., p. 151–156.

Vázquez-Padín, D. et al., 2019. Video integrity verification and GOP size estimation via generalized variation of prediction footprint. *IEEE Transactions on Information Forensics and Security.*

Verde, S. et al., 2018. *Video Codec Forensics Based on Convolutional Neural Networks.* s.l., s.n., p. 530–534.

Xia, M. et al., 2017. Detecting video frame rate up-conversion based on frame-level analysis of average texture variation. *Multimedia Tools and Applications,* Volume 76, p. 8399–8421.

Xiong, X. & la Torre, F. D., 2013. *Supervised Descent Method and Its Applications to Face Alignment.* s.l., s.n., pp. 532-539.

Yang, P. et al., 2020. Efficient video integrity analysis through container characterization. *IEEE Journal of Selected Topics in Signal Processing,* Issue 10.1109/JSTSP.2020.3008088.

Zhang, Z., 2000. A Flexible New Technique for Camera Calibration. *IEEE Trans. Pattern Anal. Mach. Intell.,* 11, Volume 22, p. 1330–1334.

Zhu, X. et al., 2015. *High-fidelity Pose and Expression Normalization for face recognition in the wild.* s.l., s.n., pp. 787-796.

Appendix A – Publications

This is the list of the publications that have been obtained during the research activity, and where an acknowledgement to the project has been added. The pdf files of these publications have all been collected in the corresponding page of our Confluence Florence space, i.e. at the link: https://mediforprogram.com/wiki/display/FLOR/Reports+and+Publications

Description	**Source**
M. Iuliani, M. Fanfani, C. Colombo, A. Piva, "Reliability Assessment of Principal Point Estimates for Forensic Applications", Journal of Visual Communication and Image Representation, Vol. 42, January 2017, Pages 65-77, ISSN 1047-3203, http://dx.doi.org/10.1016/j.jvcir.2016.11.010	2017-JVCI_PP.pdf
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 Transactions on Information Forensics and Security*, vol. 14, no. 3, March 2019, pp. 635-645, doi 10.1109/TIFS.2018.2859760	UnsupervisedAnalysisOfMP4-likeFileContainer.pdf
O. Al Shaya , P. Yang , R. Ni, Y. Zhao , and A. Piva, "A new dataset for source identification of High Dynamic Range images", *Sensors*, 2018, 18(11), 3801; https://doi.org/10.3390/s18113801.	2018_SENSORS_HDR.pdf
M. Iuliani, M. Fontani, D. Shullani, and A. Piva, "Hybrid reference-based Video Source Identification". *Sensors*, 2019, 19(3), 649, https://doi.org/10.3390/s19030649.	2019_SENSORS_HYBRID.pdf
B. Hadwiger, D. Baracchi, A. Piva and C. Riess, "Towards Learned Color Representations for Image Splicing Detection," *Proceedings of ICASSP 2019*, Brighton, United Kingdom, 2019, pp. 8281-8285.	2019_ICASSP19.pdf
M. Fanfani, F. Bellavia, M. Iuliani, A. Piva and C. Colombo, "FISH: Face Intensity-Shape Histogram representation for automatic face splicing detection". *Journal of Visual Communication and Image Representation*, 63, 102586:1-8, Elsevier 2019.	JVCI2019.pdf
M. Fanfani, M. Iuliani, F. Bellavia, C. Colombo and A. Piva, "A vision-based fully automated approach to robust	SPIC2019.pdf

Description	**Source**
image cropping detection". *Signal Processing: Image Communication*, Elsevier 2019.	
F. Bellavia, M. Iuliani, M. Fanfani, C. Colombo and A. Piva, "PRNU pattern alignment for images and videos based on scene content". In *Proceedings 26th International Conference on Image Processing ICIP* 2019, Taipei, Taiwan, September 2019, IEEE 2019.	ICIP2019.pdf
D. Vázquez-Padín, M. Fontani, D. Shullani, F. Pérez-González, A. Piva and M. Barni, "Video Integrity Verification and GOP Size Estimation via Generalized Variation of Prediction Footprint", in *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 1815-1830, 2020., doi 10.1109/ TIFS.2019.2951313	2020-TIFS_GVPF.pdf
P. Yang, D. Baracchi, M. Iuliani, D. Shullani, R. Ni, Y. Zhao, and A. Piva, "Efficient video integrity analysis through container characterization", *IEEE Journal of Selected Topics in Signal Processing* , Special Issue on Data Driven Media Authentication and Forensics, 2020, doi: 10.1109/JSTSP.2020.3008088	decision-trees-paper-rev.pdf
S. Mandelli, F. Argenti, P. Bestagini, M. Iuliani, A. Piva, S. Tubaro, "A Modified Fourier-Mellin Approach for Source Device Identification on Stabilized Videos", 27th International Conference on Image Processing ICIP 2020	20200207103657_805689_2558.pdf

List of Symbols, Abbreviations, and Acronyms

3DMM
3D Morphable Model ... 9

AUC
Area Under (the ROC) Curve ... iv; 1; 9; 10; 14; 15; 16; 19; 20

CART
Classification And Regression Trees ... 13

CC
Center of the image ... 5; 6

CNN
Convolutional Neural Network ... 26; 36

DV
Digital Video ... 22; 27

EIS
Electronic Image Stabilization ... iii; 2; 21; 24; 25; 29; 30; 31; 32; 33

FENCE
Forensic modulEs for the automated aNalysis of physiCal integrity in imagEs ... 1

FISH
Face Intensity-Shape Histogram ... 7; 8; 9; 10; 38; 43

HQ
High Quality ... 27

HSI
Hybrid Camera Identification ... iii; 22; 23

LQ
Low Quality ... 27

LUT
Look-up Tables ... 33

PCE
Peak-to-Correlation-Energy ... 21; 22; 23; 24; 25; 26; 28; 29; 30; 33; 34; 35

PP
(Camera) Principal Point ... 4; 5; 6; 43

PRNU
Photo Response Non Uniformity ... iv; 1; 2; 21; 22; 24; 25; 26; 29; 30; 31; 32; 33; 34; 40; 44

RANSAC
RANndom SAmple Consensus ... 24; 25

RGB
Red Green Blue (color space) ... 8

ROC
Receiver Operating Characteristic ... 9

sGLOH2
shifting Gradient Location and Orientation Histogram ver. 2 ... 24

SIFT
Scale-Invariant Feature Transform ... 24

VP
Vanishing Point ... 5; 6

XML
Extensible Markup Language ... 12