IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 3, 2022, accepted March 17, 2022. Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2022.3161836

Amplitude SAR Imagery Splicing Localization

EDOARDO DANIELE CANNAS, (Graduate Student Member, IEEE), NICOLÒ BONETTINI, (Graduate Student Member, IEEE), SARA MANDELLI, PAOLO BESTAGINI, (Member, IEEE), AND STEFANO TUBARO, (Senior Member, IEEE)

Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, 20133 Milan, Italy

Corresponding author: Edoardo Daniele Cannas (edoardodaniele.cannas@polimi.it)

This material is based on research sponsored by the Defense Advanced Research Projects Agency (DARPA) and the Air Force Research Laboratory (AFRL) under agreement number FA8750-20-2-1004. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA and AFRL or the U.S. Government. This work was supported by the PREMIER project, funded by the Italian Ministry of Education, University, and Research within the PRIN 2017 program. Hardware support was generously provided by NVIDIA Corporation.

ABSTRACT Synthetic Aperture Radar (SAR) images are a valuable asset for a wide variety of tasks. In the last few years, many websites have been offering them for free in the form of easy to manage products, favoring their widespread diffusion and research work in the SAR field. The drawback of these opportunities is that such images might be exposed to forgeries and manipulations by malicious users, raising new concerns about their integrity and trustworthiness. Up to now, the multimedia forensics literature has proposed various techniques to localize manipulations in natural photographs, but the same problem has never been investigated on SAR images. Forensics methods developed for natural photographs are not guaranteed to succeed on SAR images, as their generation pipeline is completely different from that of digital pictures. In this paper, we investigate the problem of localizing splicing attacks in amplitude SAR imagery. Our goal is to identify the pixels of an amplitude SAR image that have been copied and pasted from another image for malicious purposes, considering also that the attacker might have applied some editing to conceal this manipulation. To do so, we leverage a Convolutional Neural Network (CNN) to extract a fingerprint highlighting inconsistencies in the processing traces of the analyzed input. Then, we examine this fingerprint to produce a binary tampering mask indicating the pixel region under splicing attack. Results show that our proposed method, tailored to the nature of SAR signals, provides better performances than state-of-the-art forensic tools developed for natural images.

INDEX TERMS SAR, GRD, image splicing localization, deep learning, multimedia forensics, satellite imagery.

I. INTRODUCTION

Due to the lively development of Internet-based communication systems, the diffusion and sharing of multimedia content (i.e., digital images, videos or audio clips) have become part of our daily life. At the same time, we have become extremely acquainted in using tools for editing these objects. Doubts regarding whether the content we are enjoying is genuine or not are frequent every day. Indeed, from politics [1] to everyday life experience [2], the areas where fake media could possibly harm are many.

In this vein, multimedia forensics researchers aim at developing techniques to retrieve information about the multimedia object at hand. For instance, they are interested in verifying the integrity and trustworthiness of data, spotting manipulated multimedia content and localizing possible forgeries. Forensics researches typically tackle these problems by considering a simple principle: during the data life-cycle, various non-invertible operations are performed. Each operation leaves a peculiar trace, or footprint, that can be exploited to expose and localize a specific editing.

The main efforts of the community have been historically directed towards the analysis of digital images [3]. Many techniques have been developed to detect traces left by specific operations executed on the whole picture. Furthermore, many researches aimed at spatially localizing traces left by editing operations applied locally on the image (i.e., splicing localization). A few examples of local image splicing are the insertion of a portion of an image into another one, or the deletion of a pixel area from the sample under attack.

In addition to classical digital photographs, overhead imagery is recently becoming more accessible than before.

The associate editor coordinating the review of this manuscript and approving it for publication was Donato Impedovo.

This is probably due to the increased availability of satellites equipped with imaging sensors and the widespread diffusion of public websites [4] sharing this kind of images. This imagery represents data in a wide variety of modality, from optical (e.g., panchromatic, RGB), to thermal and Synthetic Aperture Radar (SAR) as well.

Despite the great availability of tools and techniques for analysing the integrity of natural images, the potential malicious editing of overhead images is a growing concern. Indeed, as any other type of digital imagery, overhead data can be easily manipulated through editing software suites (e.g., Photoshop, GIMP, etc.) as well as through synthetic image generation tools [5], [6], and examples of malicious modifications have been worrying the public opinion and media [7], [8].

Unfortunately, the footprints characterizing the overhead image life-cycle are different from those of digital photographs, and state-of-the-art methods suited for digital photographs are likely bound to perform poorly if blindly applied to satellite data. Therefore, developing techniques to localize potential manipulations applied to overhead imagery is becoming a task of paramount importance.

While the forensics community has started developing techniques specifically tailored to satellite data, to the best of our knowledge the problem of forgery localization on SAR imagery has never been investigated in the literature. However, since SAR products, especially those based on amplitude only, are easy to handle and modify even without specific expertise, their possible manipulation by malicious users is concerning.

This paper investigates the problem of splicing localization in amplitude SAR images. Specifically, we consider the situation in which a region of an amplitude SAR image has been substituted with another region coming from a different image, and some editing might have been applied to hinder this manipulation.

Our goal is to localize the manipulated region (i.e., performing splicing localization), providing a binary mask highlighting the manipulated area. To do so, we rely on Convolutional Neural Networks (CNNs) to first extract a fingerprint reporting information on the forensic traces found in the analyzed image. The fingerprint extraction stage is inspired by existing state-of-the-art multimedia forensic methods, but we reformulate it to best suit the context of SAR imagery. Then, we exploit the extracted fingerprint to generate a tampering mask showing which pixels have undergone splicing. We propose three different methods, one supervised approach relying on CNNs and two unsupervised approaches leveraging clustering techniques.

To validate our findings, we construct a custom dataset of spliced amplitude SAR images by applying forgeries of different size and considering various editing operations performed on the manipulated data. We compare with state-of-the-art algorithms for splicing localization on natural images, always achieving better localization performances. Our results suggest that the forensic analysis on manipulated amplitude SAR images is feasible, as long as the splicing localization is performed being aware of the distinct nature of SAR imagery with respect to natural photographs.

To summarize, the main contributions of our paper are listed in the following:

- We analyze the problem of splicing localization in amplitude SAR imagery, i.e., we propose a solution to localize regions of amplitude SAR images that have been copied and pasted from another sample to alter the original image content, considering also that the spliced region might have undergone some editing to conceal this manipulation. To the best of our knowledge, this is the first contribution on the matter proposed in the literature;
- The proposed solution is tailored to amplitude SAR images and has been developed analyzing in detail the life-cycle of SAR data;
- We demonstrate the viability of the forensic analysis of amplitude SAR imagery, with the proposed method reaching better performances than state-of-the-art techniques developed for natural photographs.

The rest of the paper is organized as follows. In Section II we present an overview of forensics methods developed for both natural and overhead imagery. In Section III, we provide some useful background on the deep learning tools employed in our proposed splicing localization pipeline, and on the SAR imagery generation process. In Section IV, we formulate the splicing localization problem on amplitude SAR images. In Section V, we illustrate our proposed method in details. In Section VI, we provide all the information regarding the setup used for our experiments. In Section VII, we discuss our experimental findings. Finally, in Section VIII, we draw the final considerations on our work.

II. RELATED WORKS

The presence of peer-to-peer file sharing systems in the early ’90s, and now of social media and chat services, has increased dramatically the amount of multimedia objects we enjoy everyday. At the same time, concerns regarding the genuineness of these objects have risen, pushing the multimedia forensics community to tackle the problem of verifying the integrity and trustworthiness of these data.

Historically, the main contributions focused on the analysis of digital pictures. The work by Stamm *et al.* [3] provides a detailed overview of all the techniques and tasks undertaken in the last years. For instance, different methods have been proposed to detect forensic footprints left by operations executed on the entire image. This is the case of Popescu and Farid [9], Kirchner [10], or Vázquez-Padín and Pérez-González [11], who present different techniques to expose resampling operations. Other contributions, such as Cao *et al.* [12] and Kirchner and Fridrich [13], focus on the detection of the use of median filters. The works by Bianchi and Piva [14], Thai *et al.* [15] and Mandelli *et al.*

[16] try instead to identify the execution of multiple image compressions.

Always regarding digital picture analysis, another line of research thoroughly explored is the localization of splicing attacks. Splicing refers to the insertion of a portion of an image into another one, with the possible execution of further editing, aiming at the concealment of a specific pixel area. Splicing localization means spatially identifying (i.e., at a pixel level) which areas of an image have been attacked. To spot the tampering traces, many works in the literature such as Lyu *et al.* [17], Cozzolino *et al.* [18] and Cozzolino and Verdoliva [19], rely on the information carried by the so-called noise residual. This is a picture obtained by removing the high-level semantic content from the image, for instance through high-pass filtering.

In the last years, thanks to the automatic extraction of forensic traces executed by data-driven methods, techniques coming from the deep learning area have gained a lot of popularity in the forensics field. Especially CNNs have been exhaustively explored for the task of image tampering localization. For instance, Bondi *et al.* [20] propose a framework for image splicing localization by exploiting CNN-based descriptors developed for camera model identification. Interestingly, some works combine CNNs with the idea of noise residuals: Rao and Ni [21], as well as Liu *et al.* [22], suppress the high-level image content by using a fixed high-pass convolutional filter, while Bayar and Stamm [23], [24] adopt the same approach but relying on a learned filter. More recent contributions further elaborate on this idea, providing tools that greatly improved the state-of-the-art performances on the splicing localization task. A notable example is the Noiseprint by Cozzolino and Verdoliva [25].

Due to the sensible difference in the footprints characterizing the life cycle of overhead imagery, the forensics community has developed techniques specifically tailored to satellite data, as state-of-the-art methods suited for digital photographs are likely bound to perform poorly if blindly applied to them. In this vein, Ho *et al.* [26] propose a method based on a watermarking technique to detect doctored image regions in overhead imagery. Yarlagadda *et al.* [27] show a tool for the localization of general overhead image manipulation combining a Generative Adversarial Network (GAN) with a one-class Support Vector Machine (SVM). Bartusiak *et al.* [28] rely on GANs as well for detecting and localizing RGB image forgeries. Horváth *et al.* [29] formulate the forgery detection problem on RGB data as an anomaly detection one. Similarly, Mas-Montserrat *et al.* [30] localize splicing attacks as deviations of the image pixel values from pristine distributions using generative auto-regressive models. Horváth *et al.* [31] rely on VisionTransformers [32] to build an autoencoder and localize splicing attacks as deviation from the learned latent-space distribution of pristine images. Moving to panchromatic images, Cannas *et al.* [33] localize splicing attacks relying on an ensemble of CNNs trained for sensor attribution tasks.

III. BACKGROUND

Despite the great effort put by the multimedia forensics community, the problem of splicing localization on SAR imagery has never been studied in the literature. To facilitate the discussion on the forensic analysis of amplitude SAR imagery, this section provides the reader with some useful background on SAR imaging and on the deep learning tools employed in the proposed solution.

A. DEEP LEARNING TOOLS

Deep learning is a prosperous study field that greatly improved state-of-the-art solutions for a wide range of applications, including multimedia forensics [34] as well as SAR image classification [35] and automatic target recognition [36]. Among the most used deep learning tools, CNNs had a great success as they proved handy in managing data with an intrinsic regular grid structure [37]. This is the case of digital images as well as remote sensing data, which are stored as multi-dimensional arrays.

In a nutshell, a CNN can be seen as an operator that applies a series of parametric functions (e.g., linear filtering, nonlinear saturation, matrix multiplications, etc.) to its input, in order to obtain a processed output. The parameters of the applied functions are optimized (or ‘‘learned’’) during a preliminary stage called ‘‘training’’. Depending on the task, the output of the CNN can be a label (e.g., for classification problems), a heatmap (e.g., for segmentation or localization tasks), or any other processed version of the input (e.g., for denoising purpose).

CNNs therefore allow a processing of SAR data which is adherent to their semantic. In the following sections, we illustrate some applications of these tools related to the use we make of them in our proposed method.

1) DENOISING FORENSICS

CNNs are being more and more exploited in the forensics field in the last few years [38]. In this work, we are particularly interested in the use of the Denoising Convolutional Neural Network (DnCNN) [39], a CNN developed for image denoising that has been successfully exploited for forensics tasks [25], [40]. For instance, Cozzolino and Verdoliva proposed to use a DnCNN to extract the so-called Noiseprint [25]. This is a noise-like pattern that suppresses the vast majority of the image content and exposes editing-related artifacts due to local image forgery. To extract the Noiseprint, the authors employ a particular training procedure which can be roughly summarized as follows:

1) Consider a dataset of images coming from different devices.
2) Apply the DnCNN to patches extracted from different images to obtain a series of noise patterns.
3) Keep training the DnCNN to extract similar patterns for patches coming from the same pixel region (e.g., top-left, bottom-right, etc.) of the same device,

and different patterns from patches coming from different regions and/or devices.

The last constraint is motivated by the idea of exploiting the spatial periodicity of camera-related artifacts, so that operations like image shift or rotation can be easily detected [25]. In the end, the trained DnCNN is able to extract a noise-like heatmap that, when analyzing pristine images, is self-consistent, whereas in case of spliced images clearly highlights the edited regions. This solution achieves state-of-the-art results on many image forensics datasets, and also proves useful in the analysis of remote sensing images, in particular overhead RGB data [25].

2) SEGMENTATION FORENSICS

A wide variety of CNNs-based methods have been proposed for the task of image segmentation [41]. A notable example due to its simplicity and accuracy is that of the U-Net by Ronnenberg *et al.* [42]. This network is characterized by a ''U'' shaped architecture. This is made by a contracting path, i.e., a series of convolutional layers each one followed by pooling operations, and an expanding path specular to the contracting one but containing upsampling operator rather than pooling. Skip connections are employed to concatenate and combine the output of the contracting path layers with the input of the mirrored layers of the expanding path.

In addition to image segmentation, the U-Net has found a general appreciation also in the forensics field. Kniaz *et al.* [43] used a U-Net-like architecture to train a GAN to hinder tampering artifacts in spliced images. Bi *et al.* [44] combined the idea of residual connections and U-Net architecture to realize an end-to-end trainable network able not only to detect image manipulation attacks, but also to localize them precisely.

B. SAR IMAGING

SAR imagery has been widely adopted for a variety of tasks thanks to its characteristics of providing high-resolution images independently from cloud coverage, weather conditions and daylight [45]–[47]. Earth monitoring, 2D and 3D Earth surface mapping and change detection are just few examples of successful exploitation of SAR data [48].

A SAR system is an imaging radar mounted on a platform moving in one direction (e.g., a satellite, an aircraft, etc.). While moving, the system emits sequential high power electromagnetic waves through its antenna. Waves interact with the objects they hit (i.e., the Earth surface) and are backscattered with modified amplitude and phase according to objects permittivity and physical properties (e.g., geometry, roughness). The antenna then collects these backscattered echoes that can be processed for the SAR image formation.

A simplified schematic representation of this process is provided in Figure 1. The coordinates of SAR data are related to the motion of the platform at acquisition time. As we can see in Figure 1, the first dimension corresponds to the range (or fast time), i.e., the direction perpendicular to platform flight along which the electromagnetic beam travels. The second dimension corresponds instead to the azimuth (or slow time), which is the actual trajectory of the platform.

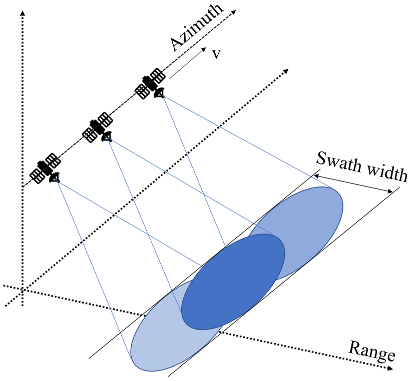


FIGURE 1. Simplified illustration of a SAR image acquisition. As the platform moves in the azimuth direction with velocity v, the antenna emits electromagnetic pulses, receiving backscattered echoes. The pulse-echo collection process senses a portion of the Earth surface (a swath) in the azimut-range coordinates.

SAR systems use frequency modulated pulse waveforms called chirps. Chirps are characterized by constant amplitude and instantaneous frequency that is linearly modulated over time. Depending on the application, different frequency bands are used for modulation, with the most popular being L (i.e., from 1 GHz to 2 GHz), C (i.e., from 3.75 GHz to 7.5 GHz) and X (i.e., from 7.5 GHz to 12 GHz) [48].

Differently from optical sensors, data coming from echo signals is not interpretable as it is. Additional processing called focusing (i.e., a double convolution both in the range and azimuth directions) is needed to obtain a visually interpretable image [48]. The resulting SAR image is a complex 2D matrix, usually displayed in terms of intensity so that pixel values approximate the reflectivity of points on the ground. This 2D matrix can be further processed, and different kinds of processing determine the existence of different so-called SAR products [47]. As an example, this can be done to ensure calibration (i.e., each pixel value represents the correct value of reflectivity) and geocoding (i.e., associate the location of each pixel with a position on the ground).

Nowadays, a wide range of SAR products can be downloaded from online platforms. Among these platforms, the Copernicus Open Access Hub [49] is the online portal provided by the European Space Agency for downloading Copernicus Sentinel-1 Mission products. The Sentinel-1 Mission products are generated according to different acquisition modes. The simplest one, the Stripmap mode, senses single continuous strips of Earth surface with a fixed antenna pattern (as Figure 1 depicts). Other acquisition modes instead acquire more than one measurement: for instance, the Interferometric Wide Swath (IW) emits three different pulses steering the antenna in the azimuth direction [50]. This operation results in the generation of three different complex images (i.e., one per pulse in the azimuth direction), or sub-swaths, provided altogether in the SAR product. SAR products usually depict very large geographic areas. Many companies allow users to

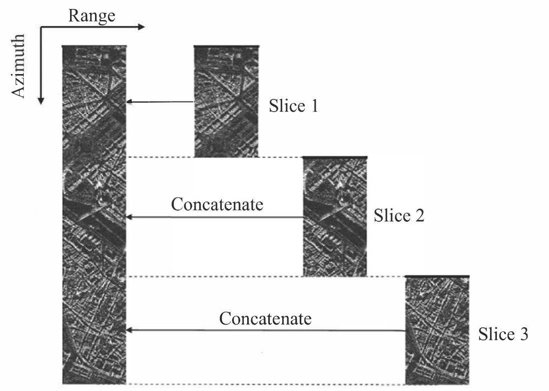


FIGURE 2. Example of product slicing for a GRD product. After each slice has been resampled to a common grid, measurements are concatenated along the azimuth direction.

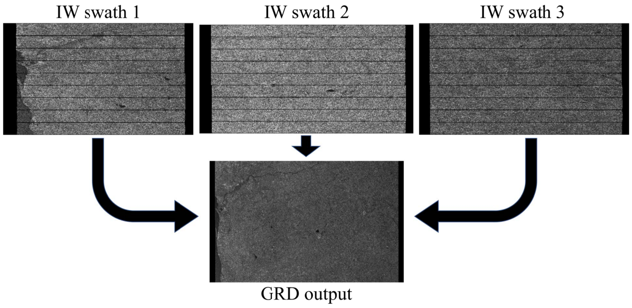


FIGURE 3. Example of sub-swaths acquired with IW, and of a full IW acquired GRD product. For each sub-swath, only the magnitude of the complex signal has been plotted. The generation of a GRD product needs a resizing of all sub-swaths in order to have them coherently merged into a single image.

select an area of interest to be imaged by their systems. To do so, as the Earth surface coverage of a single echo is often insufficient, multiple signals are collected and concatenated in a single continuous image. This is also the case of the Sentinel-1 Mission, where this operation is denoted as product slicing [51]. Figure 2 provides a graphical representation of it. Product slicing needs a resizing of all slices to a common grid in order to concatenate them.

Among the different Sentinel-1 products and acquisition modes, Ground Range Detected (GRD) products are probably the most common and accessible for a direct inspection. Indeed, they present scene reflectivity in ground-range coordinates, which are the azimuth-range coordinates projected on the Earth ellipsoid model [52]. This transformation allows to reduce the range geometric distortion and have each pixel placed in the correct position with respect to a reference plane [53]. Moreover, in case multiple sub-swaths are available, all the available signals are fused together to obtain a single continuous image. This is the case of IW acquired products. Figure 3 provides an example of such process, which is based on a resizing pipeline [54]. Finally, GRD products represent detected amplitude only, without bringing any phase information with them. All these elements make GRD products easy to handle, but also easy to be manipulated with common image editing software tools.

IV. PROBLEM FORMULATION

SAR products differ from natural photographs for a variety of reasons. For instance, the concept of single shot is hardly defined. Indeed, SAR signals are continuously acquired through moving sensors. Individual products are then generated for manageability reasons by merging several acquisitions. Moreover, some SAR products like GRD are obtained through a very specific chain of operations that has nothing in common with the ones usually employed in natural photography.

However, from the perspective of an end-user with no specific experience on overhead imagery, amplitude SAR products can be considered close to natural photographs when it comes to their manipulation. Indeed, as long as they are provided in single polarization, since they present amplitude information only they can be processed as a matrix of real numbers with any common image editing tool. This is the case of GRD products for instance.

Since GRD products, especially those acquired in IW mode, are popular [55], [56] yet easy to be manipulated, it is reasonable to consider them a vulnerable asset from a forensics perspective. Given these premises, in this work we focus on GRD products. Specifically, we consider images derived from GRD products in C-band in single vertical polarization, all acquired in IW mode. From now on, we refer to them as GRD images, or GRD tiles as they are typically mentioned in the overhead field.

In this work, we are interested in assessing the integrity of a GRD image tile at a local level and at a small granularity. In particular, given a manipulated tile, we want to localize which pixels have been affected by the editing. As manipulation we consider image splicing attacks, i.e., the insertion in a target tile of a portion coming from a different source tile. Moreover, we consider that the target region may have undergone optional editing with image processing operations (e.g., blurring, resizing, noise addition, etc.) in order to render the attack more credible and visually appealing. For instance, a resizing might be needed to match the source and target tile resolution and avoid making the splicing easily detectable at visual inspection.

More formally, we define the coordinates of a pixel of a $U \times V$ resolution tile as (u, v), where $u \in [1, \ldots, U]$ and $v \in [1, \ldots, V]$. U, V are the number of pixels per row and column, respectively. Let $\mathbf{T}_D$ and $\mathbf{T}_T$ be two pristine tiles. $\mathbf{T}_D$ is the donor tile, whereas $\mathbf{T}_T$ is the target tile. Defining $\mathcal{S}$ as the region of $\mathbf{T}_T$ under splicing attack, the resulting spliced tile $\mathbf{T}_S$ is defined as:

$$\mathbf{T}_S(u, v) = \begin{cases} e\big(\mathbf{T}_D\big)(u, v), & \text{if } (u, v) \in \mathcal{S} \\ \mathbf{T}_T(u, v), & \text{if } (u, v) \notin \mathcal{S}, \end{cases} \tag{1}$$

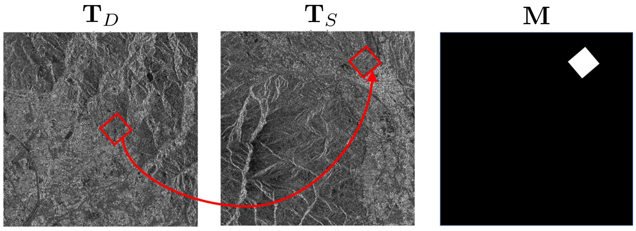


FIGURE 4. Example of a splicing operation, with donor tile $\mathbf{T}_D$ on the left, spliced tile $\mathbf{T}_S$ at the center and tampering mask $\mathbf{M}$ at the right. In the spliced tile, the outskirts of a urban area are covered.

with $e(\cdot)$ being a suitable editing function (e.g., blurring, resizing, noise addition, rotation, shearing, affine transforms, etc.).

The pixel-by-pixel integrity of the tile $\mathbf{T}_S$ can be represented by a tampering mask $\mathbf{M}$ with the same resolution of $\mathbf{T}_S$, where each pixel takes a binary value 0 or 1 depending on the pixel being pristine or manipulated, respectively. Formally, the tampering mask $\mathbf{M}$ has pixel values equal to

$$\mathbf{M}(u, v) = \begin{cases} 1, & \text{if } (u, v) \in \mathcal{S} \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

The goal of this paper is the localization of the spliced region $\mathcal{S}$ by estimating a tampering mask $\hat{\mathbf{M}}$ as close as possible to $\mathbf{M}$ from the sole analysis of the tile $\mathbf{T}_S$. Figure 4 provides a graphical representation of the splicing operation together with a tampering mask $\mathbf{M}$.

V. AMPLITUDE SAR IMAGE SPLICING LOCALIZATION

In the forensics literature, it is well known that both the acquisition device and processing operations leave peculiar traces on digital photographs. These traces can be exploited to expose forgeries [25], [57]. As the considered amplitude SAR products undergo a wide variety of operations from their acquisition to the final production (e.g., re-sampling, de-ramping, ground-range projection, etc.), it is reasonable to assume that different products may contain different processing traces. Due to the nature itself of the SAR signal and of the non-linear operations employed, even amplitude SAR products coming from the same satellite might present different traces relative to the processing executed for generating them.

Leveraging this idea, we propose a splicing localization method that exposes and highlights inconsistencies in the analyzed spliced tile $\mathbf{T}_S$ due to the different processing that target and donor tiles have undergone, and to any editing trace left by the attacker in the splicing operation. This is done by extracting a fingerprint inspired by Noiseprint [25] from the tile under analysis. This fingerprint, which suffices in spotting at a visual inspection the spliced region $\mathcal{S}$, is then further processed to estimate a binary tampering mask $\hat{\mathbf{M}}$ as close as possible to $\mathbf{M}$.

To summarize, our splicing localization process follows a two-stage pipeline (see figure 5):

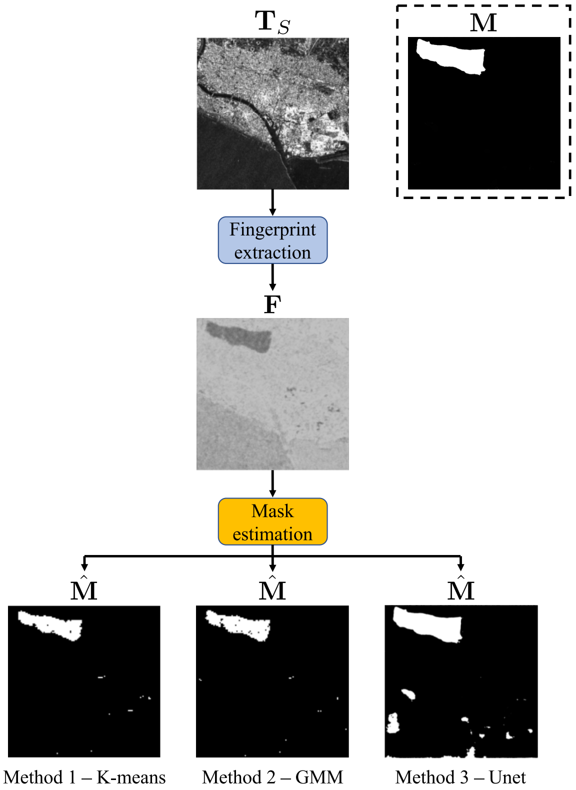


FIGURE 5. Schematic illustration of the proposed processing pipeline. A fingerprint $\mathbf{F}$ is extracted from the spliced tile $\mathbf{T}_S$ under investigation. Then, a binary tampering mask $\hat{\mathbf{M}}$ can be estimated through three different methods. For the sake of clarity, we also report the ground-truth tampering mask $\mathbf{M}$.

1) **Fingerprint extraction** - Using a properly designed CNN, a fingerprint $\mathbf{F}$ with the same resolution of $\mathbf{T}_S$ highlighting any local inconsistencies due to splicing attacks is obtained.
2) **Tampering mask estimation** - Starting from the fingerprint $\mathbf{F}$, using either unsupervised or supervised approaches, a tampering mask $\hat{\mathbf{M}}$ is estimated.

In the following, we provide additional details about each step of the proposed method.

A. FINGERPRINT EXTRACTION

The goal of this step is the extraction of a fingerprint $\mathbf{F}$ that visually highlights the spliced region $\mathcal{S}$ in an analyzed tile. To do so, we leverage the recent forensics literature. In particular, the Noiseprint [25] method shows promising results in highlighting editing traces even on data distant from natural photographs.

For our fingerprint extractor, we exploit the characterization capability offered by Noiseprint and further adapt it to the context of SAR imagery. In particular, given a spliced tile $\mathbf{T}_S$, we extract a fingerprint $\mathbf{F}$ with the same pixel resolution. Our goal is to make this fingerprint clearly highlighting spliced regions just by visual inspection. Formally, we define $\mathbf{F}$ as:

$$\mathbf{F} = f\left(\mathbf{T}_S\right), \tag{3}$$

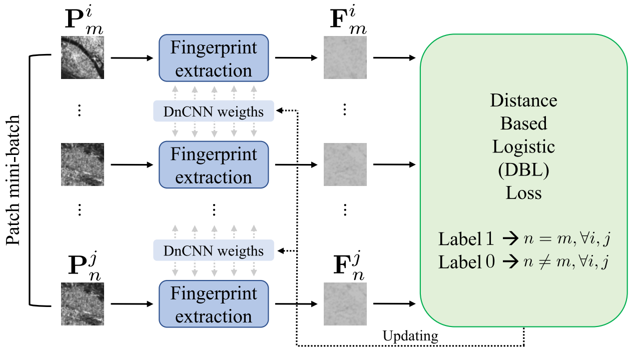


FIGURE 6. Training procedure of the fingerprint extraction block. For each patch in the mini-batch, we extract a fingerprint by means of the DnCNN. We iteratively update the DnCNN weights by comparing all the fingerprints of the mini-batch in a pair-wise fashion through the DBL loss. Pairs of fingerprints extracted from the same amplitude SAR product are associated with label 1, otherwise label 0 is assigned.

where $f(\cdot)$ represents the fingerprint extraction operator, i.e., the DnCNN network after it has been trained.

For the DnCNN training, we adopt the following pipeline:

1) We collect a number of tiles coming from M different amplitude SAR products, all generated by the same satellite. These tiles are pristine, i.e., they have not been tampered with in any way.
2) From the tiles of each product, we extract a number of patches. The i-th patch extracted from the tiles of the m-th amplitude SAR product is referred to as $\mathbf{P}_m^i$.
3) From each patch $\mathbf{P}_m^i$, we extract the related fingerprint $\mathbf{F}_m^i = f(\mathbf{P}_m^i)$.
4) We iteratively update the DnCNN weights by processing small batches of patches. In particular, given a mini-batch of patches, we process their extracted fingerprints with the Distance Based Logistic (DBL) loss presented by Võ and Hays [58]. This loss function computes the pairwise squared Euclidean distance between all the analyzed fingerprints. The objective is to make the fingerprints self-consistent if and only if they are extracted from the same amplitude product. Consistent fingerprint pairs (i.e., coming from the same SAR product) are associated with a desired low value for the Euclidean distance, while non consistent fingerprint pairs (i.e., coming from different products) are associated with a desired high Euclidean distance. More formally, we define the fingerprint pair $(\mathbf{F}_m^i, \mathbf{F}_n^j)$ as consistent if $n = m, \forall i, j$. The fingerprint pair is non consistent if $n \neq m, \forall i, j$. To do so, we assign a label 1 to all consistent pairs of fingerprints and label 0 otherwise.
5) We process the training patches by continuously updating the DnCNN weights until we reach some desired performance metrics.

For clarity's sake, Figure 6 depicts a sketch of the training pipeline of the proposed fingerprint extractor.

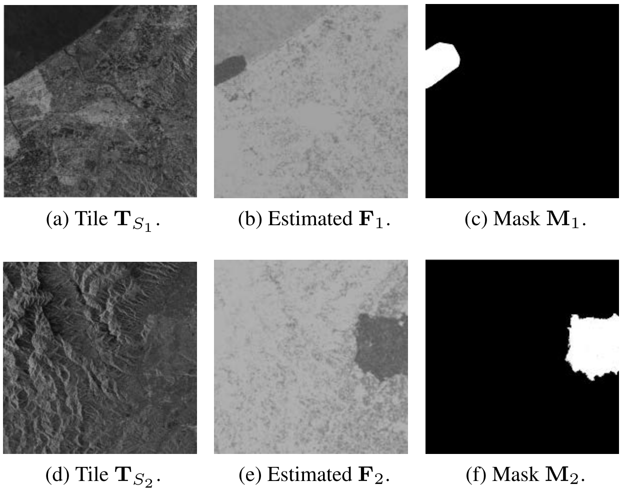
When training is finished, $f(\cdot)$ implements the desired fingerprint extraction function defined in (3). This function allows to extract a fingerprint $\mathbf{F}$ that captures traces relative to the processing pipeline of the acquired product, highlighting splicing attacks as inconsistencies in these traces. It is worth noticing that $f(\cdot)$ scales with the input resolution, so that tiles of different pixel dimensions can be processed seamlessly.

With respect to the original training procedure outlined in Section III, our pipeline has been modified taking into consideration the differences between natural images and SAR products. For instance, the presence of coherent artifacts in specific positions of the grid of pixels can be hardly demonstrated for amplitude SAR products. On top of that, pixels in natural images are temporally coherent: since they are approximately acquired at the same time instant, the signal they represent has basically no spatial nor temporal discontinuities. This is unfortunately not true for SAR products. As a matter of fact, we have seen that SAR images are generated concatenating different measurements. This implies that operations such as product splicing, together with other processing characteristic of amplitude SAR products, might alter and hinder the presence of generation artifacts with a regular spatial distribution.

We summarize our main elements of difference with respect to [25] as follows:

1) Noiseprint requires the collection of images coming from different devices (i.e., individual cameras). We exploit instead a number of tiles from M different amplitude SAR products all coming from the same satellite. Our assumption is that tiles of the same amplitude product underwent the same processing pipeline, whereas tiles of different products present different processing traces.
2) Noiseprint trains the DnCNN by comparing pairs of patches, giving a positive label in the DBL only if patches come from the same pixel region and device. Reasonably this constraint should not hold for SAR images, thus we relax it and give a positive label whenever the patches come from the same amplitude product, regardless of the pixel region of extraction.
3) Noiseprint does not employ any data-augmentation strategy during training. We propose to include resizing as data-augmentation. The reason behind this choice is twofold: on one hand, resizing might improve the extractor robustness to editing operations applied to hinder the localization of $\mathcal{S}$. On the other hand, we have seen in Section III that each SAR product is characterized by a number of resizing operations leaving peculiar traces. For this reason, we apply resizing and propose to consider all resized tiles as coming from new SAR products. This is a good solution to enlarge the number and variety of products at disposal.

Such modifications, even if at a first glance might appear negligible, will later show to improve the performances of


(a) Tile $\mathbf{T}_{S_1}$. (b) Estimated $\mathbf{F}_1$. (c) Mask $\mathbf{M}_1$.

(d) Tile $\mathbf{T}_{S_2}$. (e) Estimated $\mathbf{F}_2$. (f) Mask $\mathbf{M}_2$.

FIGURE 7. Examples of fingerprints extracted from spliced tiles. We can notice a spliced area highlighted in the fingerprints $\mathbf{F_1}$ and $\mathbf{F_2}$. The precise location of the spliced areas is reported in the tampering masks $\mathbf{M_1}$ and $\mathbf{M_2}$.

the pipeline with respect to adopting the baseline training procedure as it is.

Figure 7 reports some examples of spliced tiles, their tampering masks and the fingerprints extracted with function $f(\cdot)$. As we can see, even though the fingerprints $\mathbf{F}_1$ and $\mathbf{F}_2$ are not binary images yet, spliced areas are easily recognizable at this stage of the pipeline already.

B. TAMPERING MASK ESTIMATION

Once the fingerprint $\mathbf{F}$ is extracted, the next step in the pipeline is tampering mask estimation. This stage segments the fingerprint to generate a binary mask $\hat{\mathbf{M}}$ representing the integrity of the spliced tile $\mathbf{T}_S$ given as input.

Many forensics methods in the literature provide a binary heatmap decision highlighting the spliced region. The most common approaches rely on automatic thresholding or two-class clustering. However, such procedures may lead to a non-efficient binary partition of the feature space [59], as the presence of scene content might still appear in the generated tampering mask. Such phenomenon is noticeable also in the fingerprints shown in Figure 7. For instance, the fingerprint $\mathbf{F}_1$ presents some texture related to the sea and the urban areas surrounding $\mathcal{S}$.

Having a binary heatmap however is of paramount importance for supporting the work of forensic analysts. For this reason the second step of our pipeline is dedicated in providing the most detailed possible tampering mask $\hat{\mathbf{M}}$. To do so, we propose different methods that deeply analyze the fingerprint $\mathbf{F}$ to provide an efficient binary partition of it. To this end, we have considered three different techniques, which can be divided into two families:

- **Unsupervised** approaches. With these techniques, we first partition $\mathbf{F}$ into different clusters. Then, starting from these partitions, different candidate masks are compared to choose the most appropriate one. We propose two unsupervised methods: the first one is based on the K-means clustering algorithm [60]; the second one is based on Gaussian Mixture Models (GMMs) [61]. As unsupervised methods, both techniques do not require a preliminary stage of training.
- **Supervised** approaches. In this scenario, we estimate the tampering mask $\hat{\mathbf{M}}$ using classic CNNs adopted in the image segmentation field. Specifically, we propose a supervised strategy based on the well-known U-Net architecture [42]. This method requires a preliminary stage of training.

1) UNSUPERVISED K-MEANS-BASED MASK ESTIMATION

The K-means algorithm [60] is well known in the signal processing community and it has been historically used to perform clustering operations. Given a number of observations, the algorithm partitions them into a finite set of groups, called clusters, assigning each observation to the group showing the nearest mean distance (e.g., the Euclidean distance) from it.

The assumption behind its use in our pipeline is that the spliced region $\mathcal{S}$ is well localized in the fingerprint $\mathbf{F}$. A good estimate of the tampering mask will reasonably present a well localized cluster of pixels. In a nutshell, we propose to look for different clusters of pixels in $\mathbf{F}$, compare their compactness and then choose the most compact one to estimate the final tampering mask $\hat{\mathbf{M}}$.

To do so, we first divide $\mathbf{F}$ into a set of non-overlapping patches $\mathbf{P}_n$, $n = 1, \dots, N$, with N being the total number of patches in $\mathbf{F}$. These patches are the observations used by the K-means algorithm to cluster $\mathbf{F}$ based on their Euclidean distances. After the algorithm converges, the fingerprint $\mathbf{F}$ is divided into C clusters. We define the set of coordinates of pixels belonging to the c-th cluster as $\mathcal{P}_c = [\mathcal{U}_c, \mathcal{V}_c]$. $\mathcal{U}_c$ and $\mathcal{V}_c$ are the sets of row and column coordinates, respectively.

It is worth noticing that, if the pixels belonging to a cluster are close to each other, this might be indicative of the presence of a well localized spliced area in the fingerprint $\mathbf{F}$. We therefore need a measure of proximity of the pixels. As a metric, we propose to compute the variance of the coordinate values of the pixels belonging to each cluster. The smaller the variance, the more compact the cluster. Thus, the best localized cluster can be estimated as:

$$\hat{c} = \arg\min_{c} \mu\left(\sigma^2(\mathcal{U}_c), \sigma^2(\mathcal{V}_c)\right), \quad c = 1, \dots, C, \tag{4}$$

where σ^2 is the variance and μ is the arithmetic mean.

Starting from the pixel coordinates of $\mathcal{P}_{\hat{c}} = [\mathcal{U}_{\hat{c}}, \mathcal{V}_{\hat{c}}]$, i.e., the coordinates of the best localized cluster, we finally create a binary segmentation mask of the fingerprint $\mathbf{F}$. We do so by assigning a positive label to all the pixels belonging to that cluster, and a negative label to all those not belonging to it. Following the convention introduced in (2), we assign 1 as positive label and 0 as negative one. The final tampering mask $\hat{\mathbf{M}}$ can be formally defined as:

$$\hat{\mathbf{M}}(u, v) = \begin{cases} 1, & \text{if } (u, v) \in \mathcal{P}_{\hat{c}} \\ 0, & \text{otherwise} \end{cases}. \tag{5}$$

2) UNSUPERVISED GMM-BASED MASK ESTIMATION

Mixture distributions are a powerful statistical tool consisting in the description of data by linearly combining basic distributions, such as Bernoulli, Dirilichet, or Gaussian [61]. They have been studied for years also for the task of data clustering, and especially GMMs have proven extremely handy and simple to use adopting the Expectation Maximization (EM) algorithm [62].

The basic functioning of the EM algorithm is not too dissimilar from the K-means. Given a set of observations, the EM fits a mixture of C different Gaussian distributions to the data. The mixture is such that each c-th component groups together observations that have been likely generated by the same Gaussian distribution. This performs a clustering of the data based on the probability of each cluster having generated a data point.

Inside our pipeline, the use of GMMs is really close to K-means. In this case as well, we propose to divide the fingerprint $\mathbf{F}$ into non-overlapping patches $\mathbf{P}_n, n = 1, \dots, N$, with N being the total number of patches in $\mathbf{F}$. These are the observations used by the EM algorithm. After the algorithm converges, the fingerprint $\mathbf{F}$ is divided again into C clusters, where the elements of the clusters are paired based on how well their values can be described by the same Gaussian distribution. We then look for the most compact cluster $\hat{c}$, following the same methodology applied for the K-means mask generation method. Using (4), we look for the cluster whose pixels' coordinate values show the smallest variance. The final estimated tampering mask is defined as in (5).

3) SUPERVISED CNN-BASED MASK ESTIMATION

We propose a supervised mask estimation strategy relying on the U-Net architecture [42]. Our choice fell over this network as it is easy and fast to train, while achieving really competitive performances in the segmentation of a variety of imagery data, from SAR [63], to overhead RGB for road extraction [64], to seismic images salt segmentation [65], and of course medical imagery [66].

In the context of our proposed method, the use of the U-Net translates in using the network to generate a tampering mask estimate $\hat{\mathbf{M}}$ starting from an input fingerprint $\mathbf{F}$. The proposed method first extracts a probability mask $\hat{\mathbf{M}}^u$ defined as

$$\hat{\mathbf{M}}^u = u(\mathbf{F}), \tag{6}$$

where $u(\cdot)$ is the fingerprint segmentation function implemented by the U-Net. $\hat{\mathbf{M}}^u$ has the same pixel resolution of $\mathbf{F}$, and each pixel presents values close to 0 when there is a low probability for that pixel of being spliced, and values close to 1 when there is an high probability of manipulation.

In order for the function $u(\cdot)$ to correctly implement a coherent segmentation, the deployment of the U-Net needs a stage of training, in which the network learns to retain only the information regarding the localization of the spliced region $\mathcal{S}$. Such training can be done with a dataset of K spliced tiles $\mathbf{T}_{\mathcal{S}_k}$, $k = 1, \dots, K$. For each of them, a corresponding ground-truth tampering mask $\mathbf{M}_k$ together with an extracted fingerprint $\mathbf{F}_k$ needs to be provided.

During the training phase, every fingerprint $\mathbf{F}_k$ is processed according to (6) to obtain the tampering mask estimate $\hat{\mathbf{M}}^u_k$. The network performances are then evaluated comparing the ground-truth tampering masks $\mathbf{M}_k$ and the U-Net-estimates $\hat{\mathbf{M}}^u_k$. We propose to do so by minimizing the sum of Dice loss [67] and Focal loss [68]. For a complete definition and a more comprehensive discussion on both losses we refer the reader to the original papers. For the sake of our discussion, it suffices to say that both have proven to help reducing the overly thick boundaries in the segmented objects that usually present when training segmentation networks with a simple binary cross-entropy loss [67].

At deployment stage, we propose to impose a threshold τ on the pixel values of the estimated mask $\hat{\mathbf{M}}^u$ derived from a query fingerprint $\mathbf{F}$. The final estimated mask is equal to:

$$\hat{\mathbf{M}}(u, v) = \begin{cases} 1, & \text{if } \hat{\mathbf{M}}^u(u, v) \geq \tau \\ 0, & \text{otherwise} \end{cases}. \tag{7}$$

It is worth noticing that the training of the fingerprint extractor (presented in Section V-A) and the training of the U-Net for the forgery mask estimation do not simultaneously happen: we first need to train the fingerprint extractor to generate the fingerprint $\mathbf{F}$, and then, in a second stage, the U-Net. Furthermore, notice that the fingerprint extractor is trained on pristine tiles only; the U-Net instead needs to be trained on spliced tiles.

VI. EXPERIMENTAL SETUP

In this section, we describe the details regarding our experimental setup, including the dataset collection procedure, the training strategy together with the hyperparameters for the CNN-fingerprint extractor and mask estimation methods, and finally the metrics used for our method evaluation.

A. DATASET

As introduced in Section V, in our work we considered SAR GRD products in single vertical polarisation. More specifically, we downloaded from the Copernicus Open Access Hub 20 products acquired in IW mode coming from the Sentinel-1 mission. All products have been sensed by the same satellite (S1-B), present high spatial resolution, and overall dimensions in pixels roughly around 20000×20000.

Given the size of these acquisitions, each of them has been divided into non-overlapping tiles $\mathbf{T}$ 1024×1024 pixels wide. These operation allowed us to work at a local level with small granularity and making the input easily processable by our networks. From each product, we extracted $300 - 400$ tiles, resulting in a total of approximately 8000 tiles. These data constituted the basis for all the steps of our experiments. Indeed, starting from these samples we managed to create the following datasets:

TABLE 1. Dataset composition and use of each set of tiles.

Set	# of tiles		Training		Testing
			Fingerprint	*U-Net*	
FED	Pristine tiles	4000	✓		
SD1	Inter-spliced tiles	1600		✓	
SD2	Inter-spliced tiles	3500			✓
	Intra-spliced tiles	3500			✓

- **Fingerprint Extraction Dataset (FED)**. This is the dataset of pristine tiles used for training the fingerprint extraction function $f(\cdot)$ defined in Section V-A;
- **Spliced Dataset 1 (SD1)**. This is a dataset of spliced tiles used for training the U-Net segmentation function $u(\cdot)$ defined in Section V-B;
- **Spliced Dataset 2 (SD2)**.This dataset is again constituted by spliced samples, but that have never been seen during the training nor validation of the U-Net. We used these tiles to test the performances of the complete pipeline with all its tampering mask estimation methods.

Table 1 reports a summary of the different datasets used in the paper. In the following paragraphs, we provide further details on the creation and usage of each set.

1) FINGERPRINT EXTRACTION DATASET (FED)

For creating this dataset, we took only tiles from the first 10 SAR products we downloaded. More specifically, we took the 50% of tiles from each GRD product, reserving the remaining 50% for creating the Spliced Dataset 1 (SD1). In this way we assured that the training of the U-Net happened on samples never seen during training by the fingerprint extractor. In the end, we created a dataset of approximately 4000 pristine tiles for training the fingerprint extractor.

2) SPLICED DATASET 1 (SD1)

The SD1 is a dataset of splicing attacks created for training the U-Net. The tiles have been taken from the first 10 GRD products downloaded for our experiments. More specifically, we took the 50% of tiles not used for training the fingerprint extractor.

The splicing attacks have been realized in four different scenarios:

1) with the donor tile $\mathbf{T}_D$ having undergone no editing;
2) with the donor tile $\mathbf{T}_D$ having undergone a rotation with angle chosen randomly;
3) with the donor tile $\mathbf{T}_D$ having undergone resizing;
4) with the donor tile $\mathbf{T}_D$ having undergone both rotation and resizing.

For all four scenarios, we always considered the case where the donor tile $\mathbf{T}_D$ and the target tile $\mathbf{T}_T$ come from different products. We generated spliced tiles $\mathbf{T}_S$ with a spliced region $\mathcal{S}$ contained inside a 128×128 or 256×256 pixel area. More specifically, we proceeded as follows:

1) we applied a selected editing operation to $\mathbf{T}_D$;
2) we randomly cropped a pixel region from $\mathbf{T}_D$, imposing it to have a maximum resolution of either 128×128 or 256×256 pixels;
3) we selected a random position in the target tile $\mathbf{T}_D$ and pasted the spliced region $\mathcal{S}$ on it.

We considered different combination of parameters for the editing, resulting in a final number of 1600 samples. For clarity's sake, Table 2 reports all the considered editing operations with their parameters.

3) SPLICED DATASET 2 (SD2)

The SD2 is a second dataset of splicing attacks designed to test the performance of the complete pipeline. The composition of the SD2 has been executed starting from the tiles of the last 10 GRD products at our disposal. These products have been reserved to this task to avoid any possible overlap between the data used for training the data-driven components of our pipeline, and the data used for testing them.

For the generation of the SD2, we wanted a more challenging dataset with respect to the SD1. To do so, we considered both the cases where the donor tile $\mathbf{T}_D$ and the target tile $\mathbf{T}_T$ come from different or the same products. We define the first scenario as inter-splicing and the latter one as intra-splicing. Moreover, we extended the number of editing operations applied on $\mathbf{T}_D$ using processing never seen by the U-Net. For this last aspect, we tried to simulate an attacker perspective and considered operations that could make the tampering more plausible in the SAR imaging context.

We used noise addition with two different distributions (Gaussian and Laplacian), two typologies of blurring (average, median), a similarity transformation comprehending rotation and scaling, a speckle-like multiplicative noise degradation and, finally, we considered also the case where no editing is applied to $\mathbf{T}_D$. The parameters used for executing the editing are all reported in Table 2.

We also varied the dimensions of the spliced region $\mathcal{S}$. Starting from the previous maximum pixel resolutions of 128×128 and 256×256 pixels, we included the intermediate areas of 160×160, 192×192 and 224×224. In the end, 7000 tiles compose the SD2, 3500 realized in the inter-splicing scenario, and 3500 realized in the intra-splicing one. These numbers account for 100 spliced tiles per area and operation, multiplied by 2 accounting for the inter and intra-splicing modalities.

B. TRAINING

Here we briefly illustrate the training procedures followed for the data-driven components of our pipeline, i.e., the fingerprint extractor and the U-Net mask estimator.

1) FINGERPRINT EXTRACTION

The training set was constituted by the pristine tiles coming from the Fingerprint Extraction Dataset (FED) dataset. For the fingerprint extractor, we relied on the mini-batch boost procedure originally employed by Cozzolino and

TABLE 2. Parameters used for the editing operated in the SD1 and Spliced Dataset 2 (SD2) and total number of samples. Parameters for noise-based editing are reported referring to samples with values between 0 and 1.

Set	Editing operation	Editing parameters	Total	Total # in set
SD1	*No editing*		200	1600
	Random rotation	Angle $\sim \mathcal{U}(-45°, 45°)$	200	
	Resize 1	Factor $= 1.5$	200	
	Resize 2	Factor $= 2$	200	
	Resize 3	Factor $= 2.5$	200	
	Random rotation & resize 1	Angle $\sim \mathcal{U}(-45°, 45°)$; Factor $= 1.5$	200	
	Random rotation & resize 2	Angle $\sim \mathcal{U}(-45°, 45°)$; Factor $= 2$	200	
	Random rotation & resize 3	Angle $\sim \mathcal{U}(-45°, 45°)$; Factor $= 2.5$	200	
SD2	*No editing*		1000	7000
	Additive Gaussian noise	Mean $= 0$; Var $\sim \mathcal{U}(0, 0.1)$	1000	
	Additive Laplacian noise	Mean $= 0$; Var $\sim \mathcal{U}(0, 0.1)$	1000	
	Average blur	Kernel dim $= 10 \times 10$	1000	
	Median blur	Kernel dim $= 5 \times 5$	1000	
	Random rotation & resize	Angle $\sim \mathcal{U}(-45°, 45°)$; Factor $\sim \mathcal{U}(1, 1.5)$	1000	
	Speckle-like noise	Mean $= 0$; Var $\sim \mathcal{U}(0, 0.3)$	1000	

Verdoliva [25]. However, due to the limited amount of products available with respect to the original forensic task, we exploited more patches for the construction of the mini-batches. Specifically, in each batch we accounted for 4 GRD products at a time, inserting 10 tiles per product. From each tile, we randomly extracted 6 patches of 48×48 pixels, ending up with 240 patches per mini-batch.

In executing the training, we investigated three scenarios, leading to three different fingerprint extractors:

- Baseline Extractor (BE): this extractor corresponds to training the fingerprint extractor proposed in [25] off-the-shelf on amplitude SAR images. Specifically, we trained the DnCNN without relaxing the constraint on the position of the patches to compute the DBL loss (see Section V-A for details). This extractor served as a baseline to evaluate the goodness of our proposed fingerprint extraction method. We randomly selected 5 products for training and 5 for validation, corresponding to 2000 tiles for training and 2000 for validation.
- SAR Adapted Extractor (SAE): in this scenario, we relaxed the patch position constraint following the motivations reported in Section V-A. This translated into assigning a positive label in the DBL loss to every patch pair coming from tiles of the same product, regardless of the position from which the patches have been extracted. As in BE, we randomly picked 5 products for training and 5 for validation, ending up with 2000 tiles for training and 2000 for validation.
- Augmented SAR Adapted Extractor (ASAE): for this extractor, we relaxed the patch position constraint as done for the SAE, but we also applied data augmentation. More specifically, we resized all the tiles using a 1.5 scaling factor, and then randomly cropped them to 1024 × 1024 pixels. As previously explained in Section V-A, we considered all resized tiles as coming from separate GRD products. We exploited 20 pristine products, i.e., 10 original products and 10 new products corresponding to their resized versions, randomly using 10 for training (corresponding to 4000 tiles), and 10 (other 4000 tiles) for validation.

All the extractors have been trained for 500 maximum epochs, with each epoch consisting of 128 batch iterations, using Adam optimizer [69] with a learning rate of 10^{-4}. We stopped the training if the validation loss did not improve for 30 consecutive epochs. Then, we kept the model showing the best validation loss.

2) U-NET MASK ESTIMATOR

Since the goal of the mask estimation task is highlighting potential forged areas in the fingerprint extracted from the query tile, the U-Net training dataset must consist of fingerprints. Therefore, we extracted the fingerprints of the samples in the SD1 by exploiting the three extractors listed in Section VI-B1. Then, we trained a U-Net on each set of fingerprints, creating a separate pipeline for each extractor.

We relied on the U-Net model reported in [70], which allows to use various CNNs as backbones for the encoder-decoder structure. Our choice fell on the EfficientNetB0 model [71], a network of the EfficientNet family which proved extremely handy and recently found a discrete success both in multimedia forensics [72]–[74] and in overhead imagery analysis [75]. We considered an EfficientNetB0 as encoder, and another one as decoder.

We randomly split the fingerprints extracted from the SD1 dataset into 50% for training and 50% for validation. We trained the networks for 500 epochs, using as loss function the one described in Section V-B and resorting to Adam optimization with a learning rate of 10^{-4}. We reduced the learning rate by a 0.1 factor on plateau of the validation loss for 10 consecutive epochs, and early-stopped the training if the validation loss did not improve for 30 consecutive epochs. We kept the best validation model for all the networks trained with the three different fingerprint extractors.

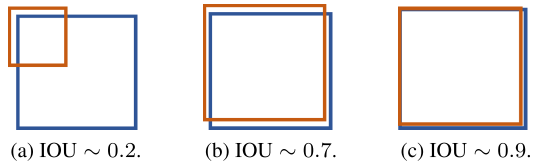


FIGURE 8. IOU localization examples. The blue square represents the ground-truth area, while the orange square shows the predicted area.

C. TAMPERING MASK ESTIMATION PARAMETERS

In order for our unsupervised methods (i.e., K-means and GMM) to be successfully deployed, the dimension of the extracted patches $\mathbf{P}_n$, as well as the number C of clusters in which the fingerprint $\mathbf{F}$ is partitioned, are crucial aspects. A too big resolution of each $\mathbf{P}_n$ or a small number of clusters C might lead to an under partition of the fingerprint, which is generally way less preferable with respect to an over segmentation.

For this reason, we spent a preliminary part of our work in determining the right amount of clusters and the right patches resolution, finding a good trade-off in dividing the fingerprint $\mathbf{F}$ into non-overlapping patches 8×8 pixels wide, and using 7 clusters. Also the U-Net needs to have a correct threshold τ applied to the probability mask $\hat{\mathbf{M}}^u$. In this case, we found an optimal value with $\tau = 0.5$.

D. EVALUATION METRICS

For evaluating our performances in correctly estimating the tampering mask, we relied on two metrics: the balanced accuracy and the Jaccard index or Intersection Over Union (IOU). Given an estimated tampering mask $\hat{\mathbf{M}}$, we can divide its pixels based on the correctness of the tampering localization. Specifically, we can assign them to four different categories:

- True Positives (TP): spliced pixels classified as spliced;
- False Positives (FP): pristine pixels classified as spliced;
- True Negatives (TN): pristine pixels classified as pristine;
- False Negatives (FN): spliced pixels classified as pristine.

The balanced accuracy is defined as:

$$\mathrm{BA} = \frac{1}{2}\left(\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} + \frac{\mathrm{TN}}{\mathrm{TN}+\mathrm{FP}}\right). \tag{8}$$

This quantity measures how well our pipeline performed in correctly assigning each pixel in the estimated tampering mask $\hat{\mathbf{M}}$, taking into account the disproportion between pristine and spliced pixels. The higher the balanced accuracy, the better the splicing localization.

The IOU is defined as:

$$\mathrm{IOU} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}+\mathrm{FN}}. \tag{9}$$

This measure is popular in computer vision for object detection tasks, where it is used to quantify how well a predicted bounding-box for an object overlaps with the actual object's position. For our task, this translates in the IOU accurately quantifying how well the area localized in the estimated tampering mask $\hat{\mathbf{M}}$ overlaps with the one indicated in the original mask $\mathbf{M}$. Figure 8 shows some examples. Values close to 1 are better, but an IOU equal or greater than 0.5 is good too.

VII. RESULTS AND ANALYSIS

In this section, we report the results of our experimental campaign. In particular, we describe the achieved results for the splicing localization task and compare our performances with state-of-the-art.

To have a fair comparison among the different fingerprint extraction and mask estimation methods, we resort to dataset SD2 for evaluating all the results. It is worth noticing that SD2 comprehends acquisitions never seen by any of the data-driven blocks of our pipeline, together with new unseen editing operations. We show our results by considering separated the two different scenarios of donor and target tiles coming from the same (i.e., intra-splicing) or different (i.e., inter-splicing) acquired products.

Tables 3 and 4 report the localization results by combining the two proposed fingerprint extractors (i.e., SAE and ASAE) and the baseline fingerprint extractor (i.e., BE). Specifically, Table 3 depicts the results achieved in the inter-splicing scenario. On the contrary, Table 4 shows results for the intra-splicing scenario. Finally, Figures 9 and 10 report some examples of splicing attacks together with all the artifacts generated by our proposed pipeline. In the following we report the major findings from these results.

A. FINGERPRINT EXTRACTORS COMPARISON

The best fingerprint extractor is always the ASAE, with the mask estimation methods GMM and U-Net alternating in providing the best performances. Moreover, the SAE showed on average better results than the baseline extractor, on 4 editing operations out of 7 in the inter-splicing scenario and on 6 operations out of 7 in the intra-splicing scenario.

Notice that, while the relaxation of the patch position constraint proposed in Section V-A (i.e., the SAE configuration) provided us better average metrics with respect to the baseline, the additional insertion of a simple data augmentation like resizing (i.e., the ASAE configuration) gave us an even greater performance boost. From this point of view, it is worth mentioning that the ASAE-based detectors showed better results also on editing operations which are not strictly related to resizing. This is true for the noise addition and blurring operations, for instance.

B. "NO EDITING": INTER-SPLICING VERSUS INTRA-SPLICING

On the "No editing" operation, all the pipelines presented the worst results. Moreover, we observed significant differences in performances depending on the donor and target tiles coming from different (i.e., inter-splicing) or the same (i.e., intra-splicing) acquired products. In the first scenario, with best IOU and balanced accuracy of 0.25 and 0.69 respectively, performances were still fairly good. In the second scenario,

TABLE 3. Average localization results per operation and per binary mask estimation method in the inter-splicing scenario. Best results in bold.

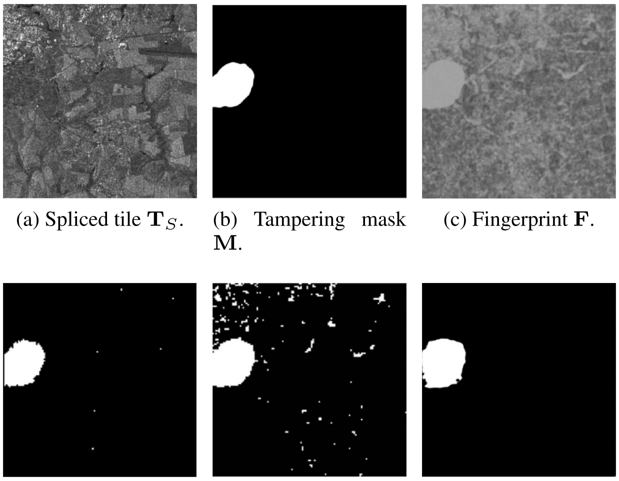
Operation	Fingerprint extractor	IOU per method			BA per method		
		GMM	*K-means*	*U-Net*	*GMM*	*K-means*	*U-Net*
No editing	BE (baseline)	0.232	0.156	0.185	0.643	0.605	0.640
	SAE	0.163	0.036	0.241	0.595	0.527	0.691
	ASAE	0.163	0.044	**0.258**	0.594	0.530	**0.697**
Additive Gaussian noise	BE (baseline)	0.462	0.243	0.297	0.787	0.687	0.772
	SAE	0.459	0.248	0.200	0.777	0.747	0.664
	ASAE	**0.669**	0.544	0.433	0.867	0.853	**0.913**
Additive Laplacian noise	BE (baseline)	0.481	0.264	0.300	0.795	0.687	0.772
	SAE	0.497	0.266	0.174	0.804	0.761	0.629
	ASAE	**0.722**	0.596	0.451	0.895	0.891	**0.914**
Average blur	BE (baseline)	0.748	0.143	0.214	0.899	0.588	0.844
	SAE	0.603	0.539	0.645	0.889	0.800	0.939
	ASAE	**0.859**	0.704	0.697	0.940	0.860	**0.972**
Median blur	BE (baseline)	0.517	0.206	0.368	0.803	0.649	0.885
	SAE	0.685	0.513	0.651	0.876	0.783	0.941
	ASAE	**0.811**	0.627	0.694	0.918	0.816	**0.972**
Random rotation & resize	BE (baseline)	0.406	0.228	0.368	0.734	0.674	0.882
	SAE	0.417	0.174	0.625	0.745	0.590	0.952
	ASAE	0.593	0.379	**0.681**	0.823	0.714	**0.972**
Speckle-like noise	BE (baseline)	0.503	0.259	0.306	0.809	0.706	0.798
	SAE	0.428	0.219	0.154	0.778	0.742	0.617
	ASAE	**0.721**	0.568	0.495	0.900	0.895	**0.942**

TABLE 4. Average localization results per operation and per binary mask estimation method in the intra-splicing scenario. Best results in bold.

Operation	Fingerprint extractor	IOU per method			BA per method		
		GMM	*K-means*	*U-Net*	*GMM*	*K-means*	*U-Net*
No editing	BE (baseline)	0.024	0.021	0.027	0.502	0.500	0.510
	SAE	0.025	0.024	0.022	0.501	0.502	0.512
	ASAE	**0.028**	0.026	0.015	0.502	0.501	**0.516**
Additive Gaussian noise	BE (baseline)	0.300	0.112	0.212	0.699	0.609	0.701
	SAE	0.420	0.191	0.093	0.756	0.706	0.557
	ASAE	**0.602**	0.482	0.390	0.834	0.837	**0.854**
Additive Laplacian noise	BE (baseline)	0.364	0.186	0.234	0.738	0.662	0.707
	SAE	0.408	0.195	0.106	0.758	0.705	0.562
	ASAE	**0.663**	0.532	0.426	0.869	0.864	**0.887**
Average blur	BE (baseline)	0.700	0.082	0.274	0.881	0.544	0.894
	SAE	0.478	0.616	0.645	0.819	0.825	0.943
	ASAE	**0.791**	0.764	0.709	0.906	0.882	**0.953**
Median blur	BE (baseline)	0.584	0.310	0.468	0.838	0.714	0.926
	SAE	0.694	0.597	0.674	0.859	0.811	**0.957**
	ASAE	**0.787**	0.684	0.717	0.899	0.842	0.956
Random rotation & resize	BE (baseline)	0.344	0.323	0.418	0.711	0.722	0.925
	SAE	0.328	0.265	0.609	0.712	0.646	0.949
	ASAE	0.543	0.497	**0.671**	0.787	0.782	**0.953**
Speckle-like noise	BE (baseline)	0.352	0.098	0.273	0.728	0.623	0.776
	SAE	0.366	0.164	0.035	0.747	0.711	0.519
	ASAE	**0.615**	0.501	0.486	0.857	0.869	**0.916**

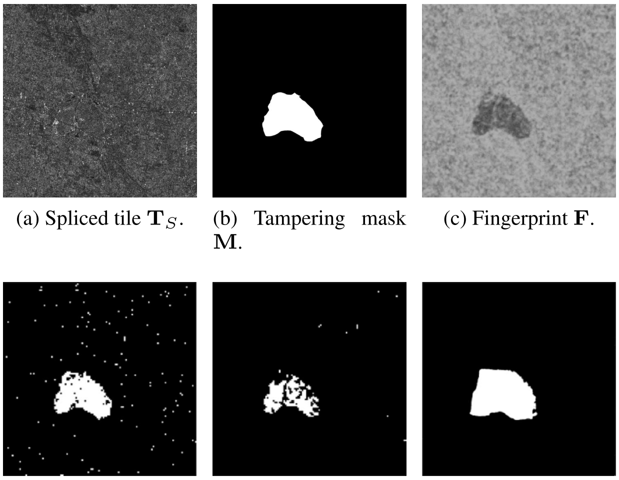
the evaluation metrics depicted instead an almost random decision for the estimation of the tampering mask.

This different behavior was somehow expected. As explained in Section V, our proposed pipeline has been designed with the objective of capturing inconsistencies related to the generation process of amplitude SAR products. The fingerprint extraction has been trained to provide a globally incoherent fingerprint only if the spliced region and its surrounding areas have undergone different processing. In intra-splicing attacks, when no editing is applied, splicing inconsistencies are absent as $\mathbf{T}_D$ and $\mathbf{T}_T$ come from the same GRD product. For this reason, we expected our detectors not being able to localize such attacks.


(a) Spliced tile $\mathbf{T}_S$. (b) Tampering mask $\mathbf{M}$. (c) Fingerprint $\mathbf{F}$.

(d) GMM estimated tampering mask $\hat{\mathbf{M}}$. (e) K-means estimated tampering mask $\hat{\mathbf{M}}$. (f) U-Net estimated tampering mask $\hat{\mathbf{M}}$.

FIGURE 9. Example of spliced tile with resizing and gaussian noise addition applied to the donor tile $\mathbf{T_D}$. The pipeline is based on the ASAE.


(a) Spliced tile $\mathbf{T}_S$. (b) Tampering mask $\mathbf{M}$. (c) Fingerprint $\mathbf{F}$.

(d) GMM estimated tampering mask $\hat{\mathbf{M}}$. (e) K-means estimated tampering mask $\hat{\mathbf{M}}$. (f) U-Net estimated tampering mask $\hat{\mathbf{M}}$.

FIGURE 10. Example of spliced tile. An urban area is covered with resizing applied to the donor tile $\mathbf{T_D}$. The pipeline used is based on the ASAE.

C. GENERALIZATION ON EDITING OPERATIONS

With the only exception of the "No editing" scenario, in Table 3, the results achieved on each editing operation always depict an IOU greater than 0.66 and a balanced accuracy exceeding 0.91. In Table 4, we exceeded 0.60 and 0.85 for IOU and balanced accuracy, respectively. While intra-splicing results are lower than inter-splicing, we expected such a behavior for the reasons reported above: spliced tiles in intra-splicing modality do not present inconsistencies in the forensic traces related to the pipeline that generated their original products, making them a more difficult asset to analyze. However, the overall good performances also in the intra-splicing scenario suggest the proposed pipeline was useful in finding inconsistencies associated to general editing operations executed on $\mathcal{S}$.

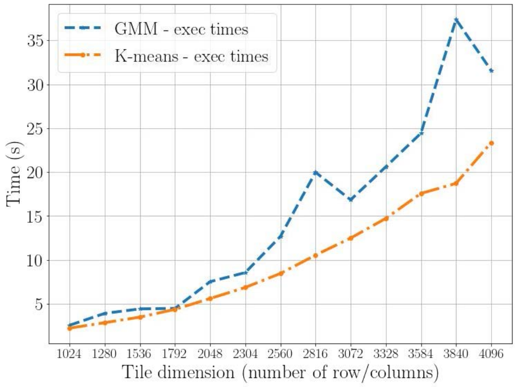


FIGURE 11. Execution times against tile resolution for the K-means and GMM-based tampering mask estimation methods.

It is interesting to notice that all the methods performed consistently across the different types of editing considered (i.e., noise addition, blurring, rotation, resizing, noise multiplication). In particular, the results achieved by the U-Net, which showed the best balanced accuracy for noise-based attacks, were quite surprising, considering that the U-Net was trained only on resizing-based attacks.

D. SUPERVISED VERSUS UNSUPERVISED APPROACHES

Comparing the different mask estimation methods, we can notice that GMM and U-Net alternated in providing the best performances. The U-Net represented the best method for all editing operations in terms of balanced accuracy, while the GMM provided best results in terms of IOU on 5 operations out of 7.

The K-means, while being the worst method of the three, showed nevertheless fairly good results. For instance, it was better than the U-Net in 5 operations out of 7 in terms of IOU in both inter and intra-splicing scenarios. Moreover, with respect to the GMM-based method, the K-means presented shorter computational times. Figure 11 reports a small performance study where we have computed, varying the resolution of the analyzed sample, the time needed by both algorithms to generate a tampering mask. We can see that for samples close to 1024 × 1024 pixels the two methods showed similar performances, but, starting from resolutions equal or greater than 2000 × 2000 pixels, K-means was considerably faster. We argue this behaviour was motivated by the EM algorithm requiring more iterations to reach similar convergence to the K-means, with each iteration requiring more computations [76]. Therefore, at deployment stage an end user might prefer to sacrifice the localization performances of the GMM method in place of the faster execution times of K-means.

Finally, while the U-Net-based strategy might seem the most promising one on average, we must be aware that, as a supervised technique, it needed a preliminary stage of training. Since the training of CNNs takes time and computational resources, the GMM and K-means methods may be appealing as well, depending on the final needs and the resources at disposal. As unsupervised methods, they are faster to deploy while still providing good performances.

TABLE 5. Best localization results per operation executed on the SD2 tiles in the inter-splicing scenario. The best tampering mask estimation method has been chosen from table 3. Best results in bold.

Operation	Best IOU per extractor				Best BA per extractor			
	BE (Noiseprint)	*SAE*	*ASAE*	*Splicebuster*	*BE (Noiseprint)*	*SAE*	*ASAE*	*Splicebuster*
No editing	0.232	0.241	**0.258**	0.194	0.643	0.691	**0.697**	0.619
Additive Gaussian noise	0.462	0.459	**0.669**	0.550	0.787	0.777	**0.913**	0.799
Additive Laplacian noise	0.481	0.497	**0.722**	0.552	0.795	0.804	**0.914**	0.796
Average blur	0.748	0.645	**0.859**	0.478	0.899	0.932	**0.972**	0.808
Median blur	0.517	0.685	**0.811**	0.508	0.885	0.941	**0.972**	0.770
Random rotation & resize	0.406	0.625	**0.681**	0.551	0.882	0.952	**0.972**	0.753
Speckle-like noise	0.503	0.428	**0.721**	0.537	0.809	0.778	**0.942**	0.781

TABLE 6. Best localization results per operation executed on the SD2 tiles in the intra-splicing scenario. The best tampering mask estimation method has been chosen from table 4. Best results in bold.

Operation	Best IOU per extractor				Best BA per extractor			
	BE (Noiseprint)	*SAE*	*ASAE*	*Splicebuster*	*BE (Noiseprint)*	*SAE*	*ASAE*	*Splicebuster*
No editing	0.027	0.025	**0.028**	0.017	0.510	0.512	**0.516**	0.481
Additive Gaussian noise	0.300	0.420	**0.602**	0.575	0.701	0.756	**0.854**	0.814
Additive Laplacian noise	0.364	0.408	**0.663**	0.599	0.738	0.758	**0.887**	0.824
Average blur	0.700	0.645	**0.791**	0.524	0.894	0.943	**0.953**	0.786
Median blur	0.584	0.694	**0.787**	0.661	0.926	**0.957**	0.956	0.770
Random rotation & resize	0.418	0.609	**0.671**	0.559	0.925	0.949	**0.953**	0.807
Speckle-like noise	0.352	0.366	**0.615**	0.606	0.776	0.747	**0.916**	0.823

E. COMPARISON WITH STATE-OF-THE-ART

For what concerns the comparison with state-of-the-art, Tables 5 and 6 summarize the best localization results achieved by the two proposed fingerprint extractors for each editing operation, along with the results obtained by the Noiseprint method [25] and the Splicebuster method by Cozzolino *et al.* [18]. We selected these techniques as they are widely exploited as a baseline in the forensics literature. Moreover, they proved to be robust to standard editing operations and do not require many adaptations to the domain of data under investigation.

Since the Noiseprint produces real-valued heatmaps (without binarization), all the tampering masks have been created starting from the extracted fingerprint and following our proposed mask estimation methods (i.e., the K-means, GMM and U-Net). The achieved results exactly correspond to the BE results that we have previously shown. Splicebuster returns real-valued heatmaps as well, but in this case we estimated binary tampering masks following the methodology suggested by the same authors in [19].

The state-of-the-art results always showed inferior performances than our best localization method. Nonetheless, especially looking at the editing operations involving noise addition or multiplication (i.e., the Speckle-like noise), Splicebuster presented even better performances than methods based on BE and SAE extractors. Despite the clear differences between natural images and amplitude SAR products, from the nature of the signals they depict to the different processing that leads to their formation, such results seem to indicate that forensics tools based on generic footprints might reveal to be useful also in the SAR context. This is especially true if the attacker relies on common editing operations (e.g., resizing, blurring, etc.), where features like the high-pass frequency co-occurrences used by Splicebuster are robust enough to be an effective splicing localization tool.

VIII. CONCLUSION

In this paper, we analyzed the problem of splicing localization in amplitude SAR imagery. The forensic analysis of these objects is becoming of paramount relevance, as amplitude SAR products are relatively easy to handle and process, even with general editing software such as GIMP or Photoshop.

To the best of our knowledge, no solution has been proposed yet in the forensics literature tailored to this kind of signals. As a matter of fact, amplitude SAR products present a completely different nature with respect to natural imagery, therefore are posing new and different challenges in assessing their integrity.

Inspired by a state-of-the-art method developed for natural images, we proposed a new splicing localization technique specifically designed for amplitude SAR products. Our proposed method extracts a fingerprint localizing spliced regions in SAR tiles. Then, the fingerprint can be analyzed using three different methods, one supervised and two unsupervised, to generate a final tampering mask. This mask is a binary heatmap indicating whether pixels underwent splicing or not. We generated different datasets of spliced GRD tiles, to train and test the validity of our proposed method. We considered different kind of manipulations applied to the tiles, from noise-based attacks to blurring and resizing.

All proposed techniques showed encouraging results in the localization of splicing attacks, providing better performances when compared to state-of-the-art solutions developed for natural images. The supervised approach reported the best numbers in terms of balanced accuracy, however needing a preliminary stage of training. The unsupervised approaches showed instead better performances in terms of the IOU metrics, and while they are less accurate in terms of balanced accuracy, they do not require a training phase.

These results proved the feasibility of the forensic analysis of amplitude SAR imagery, paving the way to further investigations on the development of methods tailored to this kind of signals. Possible research themes regard the evaluation of the proposed pipeline over elaborated splicing attacks (e.g., GAN-generated inpainting) that should in principle be detected by our technique, a specific exploitation of traces related to the generation pipeline of SAR images, the use of physics-based clues linked to the scene represented in the data and, finally, the adaptation of splicing localization methods for electrical-optical imagery.

REFERENCES

[1] S. Agarwal, H. Farid, Y. Gu, M. He, K. Nagano, and H. Li, ''Protecting world leaders against deep fakes,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2019, pp. 38–45.

[2] CNN. (Oct. 2020). *Is That Video Real?*. Accessed: Oct. 23, 2020. [Online]. Available: https://edition.cnn.com/interactive/2020/10/us/manipulated-media-tech-fake-news-trnd/

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

[4] GIS Geography. (Aug. 2017). *15 Free Satellite Imagery Data Sources*. Accessed: Jan. 1, 2022. [Online]. Available: http://gisgeography.com/free-satellite-imagery-data-list

[5] L. Abady, M. Barni, A. Garzelli, and B. Tondi, ''GAN generation of synthetic multispectral satellite images,'' *Proc. SPIE*, vol. 11533, Sep. 2020, Art. no. 115330L.

[6] B. Zhao, S. Zhang, C. Xu, Y. Sun, and C. Deng, ''Deep fake geography? When geospatial data encounter artificial intelligence,'' *Cartogr. Geograph. Inf. Sci.*, vol. 48, no. 4, pp. 338–352, Jul. 2021, doi: 10.1080/15230406.2021.1910075.

[7] BBC News. (Apr. 2016). *Conspiracy Files: Who Shot Down MH17?*. Accessed: Jan. 1, 2022. [Online]. Available: http://www.bbc.com/news/magazine-35706048

[8] Mashable. (May 2015). *Satellite Images Show Clearly That Russia Faked its MH17 Report*. Accessed: Jan. 1, 2018. [Online]. Available: http://mashable.com/2015/05/31/russia-fake-mh17-report

[9] A. C. Popescu and H. Farid, ''Exposing digital forgeries by detecting traces of resampling,'' *IEEE Trans. Signal Process.*, vol. 53, no. 2, pp. 758–767, Feb. 2005.

[10] M. Kirchner, ''Fast and reliable resampling detection by spectral analysis of fixed linear predictor residue,'' in *Proc. 10th ACM Workshop Multimedia Secur. (MM&Sec)*, 2008, pp. 11–20.

[11] D. Vázquez-Padín and F. Pérez-González, ''Prefilter design for forensic resampling estimation,'' in *Proc. IEEE Int. Workshop Inf. Forensics Secur.*, Nov. 2011, pp. 1–6.

[12] G. Cao, Y. Zhao, R. Ni, L. Yu, and H. Tian, ''Forensic detection of median filtering in digital images,'' in *Proc. IEEE Int. Conf. Multimedia Expo*, Jul. 2010, pp. 89–94.

[13] M. Kirchner and J. Fridrich, ''On detection of median filtering in digital images,'' *Proc. SPIE*, vol. 7541, Jan. 2010, Art. no. 754110.

[14] T. Bianchi and A. Piva, ''Detection of nonaligned double JPEG compression based on integer periodicity maps,'' *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 2, pp. 842–848, Apr. 2012.

[15] T. H. Thai, R. Cogranne, F. Retraint, and T.-N.-C. Doan, ''JPEG quantization step estimation and its applications to digital image forensics,'' *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 1, pp. 123–133, Jan. 2017.

[16] S. Mandelli, N. Bonettini, P. Bestagini, V. Lipari, and S. Tubaro, ''Multiple JPEG compression detection through task-driven non-negative matrix factorization,'' in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 2106–2110.

[17] S. Lyu, X. Pan, and X. Zhang, ''Exposing region splicing forgeries with blind local noise estimation,'' *Int. J. Comput. Vis.*, vol. 110, no. 2, pp. 202–221, 2014.

[18] D. Cozzolino, G. Poggi, and L. Verdoliva, ''Splicebuster: A new blind image splicing detector,'' in *Proc. IEEE Int. Workshop Inf. Forensics Secur. (WIFS)*, Nov. 2015, pp. 1–6.

[19] D. Cozzolino and L. Verdoliva, ''Single-image splicing localization through autoencoder-based anomaly detection,'' in *Proc. IEEE Int. Workshop Inf. Forensics Secur.*, Abu Dhabi, United Arab Emirates, Dec. 2016, pp. 1–6.

[20] L. Bondi, S. Lameri, D. Güera, P. Bestagini, E. J. Delp, and S. Tubaro, ''Tampering detection and localization through clustering of camera-based CNN features,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Honolulu, HI, USA, Jul. 2017, pp. 1855–1864, doi: 10.1109/CVPRW.2017.232.

[21] Y. Rao and J. Ni, ''A deep learning approach to detection of splicing and copy-move forgeries in images,'' in *Proc. IEEE Int. Workshop Inf. Forensics Secur. (WIFS)*, Dec. 2016, pp. 1–6.

[22] Y. Liu, Q. Guan, X. Zhao, and Y. Cao, ''Image forgery localization based on multi-scale convolutional neural networks,'' in *Proc. 6th ACM Workshop Inf. Hiding Multimedia Secur.*, Jun. 2018, pp. 85–90.

[23] B. Bayar and M. C. Stamm, ''A deep learning approach to universal image manipulation detection using a new convolutional layer,'' in *Proc. ACM Workshop Inf. Hiding Multimedia Secur.*, Vigo, Spain, Jun. 2016, pp. 5–10.

[24] B. Bayar and M. C. Stamm, ''Design principles of convolutional neural networks for multimedia forensics,'' in *Proc. Int. Symp. Electron. Imag., Media Watermarking, Secur., Forensics (IS&T)*, 2017, pp. 77–86.

[25] D. Cozzolino and L. Verdoliva, ''Noiseprint: A CNN-based camera model fingerprint,'' *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 144–159, 2020.

[26] A. T. S. Ho, X. Zhu, and W. M. Woon, ''A semi-fragile pinned sine transform watermarking system for content authentication of satellite images,'' in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Seoul, South Korea, Jan. 2005, p. 4.

[27] S. K. Yarlagadda, D. Güera, P. Bestagini, F. M. Zhu, S. Tubaro, and E. J. Delp, ''Satellite image forgery detection and localization using GAN and one-class classifier,'' *Electron. Imag.*, vol. 30, no. 7, pp. 214-1–214-9, Jan. 2018.

[28] E. R. Bartusiak, S. K. Yarlagadda, D. Güera, P. Bestagini, S. Tubaro, F. M. Zhu, and E. J. Delp, ''Splicing detection and localization in satellite imagery using conditional GANs,'' in *Proc. IEEE Conf. Multimedia Inf. Process. Retr. (MIPR)*, Mar. 2019, pp. 91–96.

[29] J. Horvath, D. Güera, S. K. Yarlagadda, P. Bestagini, F. M. Zhu, S. Tubaro, and E. J. Delp, ''Anomaly-based manipulation detection in satellite images,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshop (CVPRW)*, Jun. 2019, pp. 62–71.

[30] D. M. Montserrat, J. Horvath, S. K. Yarlagadda, F. Zhu, and E. J. Delp, ''Generative autoregressive ensembles for satellite imagery manipulation detection,'' in *Proc. IEEE Int. Workshop Inf. Forensics Secur. (WIFS)*, Dec. 2020, pp. 1–6.

[31] J. Horvath, S. Baireddy, H. Hao, D. M. Montserrat, and E. J. Delp, ''Manipulation detection in satellite images using vision transformer,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2021, pp. 1032–1041.

[32] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, ''An image is worth 16 × 16 words: Transformers for image recognition at scale,'' 2020, *arXiv:2010.11929*.

[33] E. D. Cannas, J. Horváth, S. Baireddy, P. Bestagini, E. J. Delp, and S. Tubaro, ''Panchromatic imagery copy-paste localization through data-driven sensor attribution,'' in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2022.

[34] L. Verdoliva, ''Deep learning in multimedia forensics,'' in *Proc. 6th ACM Workshop Inf. Hiding Multimedia Secur.*, Jun. 2018, p. 3.

[35] J. Geng, J. Fan, H. Wang, X. Ma, B. Li, and F. Chen, ''High-resolution SAR image classification via deep convolutional autoencoders,'' *IEEE Trans. Geosci. Remote Sens.*, vol. 12, no. 11, pp. 2351–2355, Nov. 2015.

[36] S. Chen, H. Wang, F. Xu, and Y.-Q. Jin, ''Target classification using the deep convolutional networks for SAR images,'' *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 8, pp. 4806–4817, Aug. 2016.

[37] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[38] I. Amerini, A. Anagnostopoulos, L. Maiano, and L. R. Celsi, "Deep learning for multimedia forensics," *Found. Trends Comput. Graph. Vis.*, vol. 12, no. 4, pp. 309–457, 2021.

[39] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, Jul. 2017.

[40] N. Bonettini, L. Bondi, D. Güera, S. Mandelli, P. Bestagini, S. Tubaro, and E. J. Delp, "Fooling PRNU-based detectors through convolutional neural networks," in *Proc. 26th Eur. Signal Process. Conf. (EUSIPCO)*, Sep. 2018, pp. 957–961.

[41] S. Minaee, Y. Y. Boykov, F. Porikli, A. J. Plaza, N. Kehtarnavaz, and D. Terzopoulos, "Image segmentation using deep learning: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Feb. 17, 2021, doi: 10.1109/TPAMI.2021.3059968.

[42] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015*. Berlin, Germany: Springer, 2015.

[43] V. Kniaz, V. Knyaz, and F. Remondino, "The point where reality meets fantasy: Mixed adversarial generators for image splice detection," in *Proc. NeurIPS*, 2019, pp. 1–12.

[44] X. Bi, Y. Wei, B. Xiao, and W. Li, "RRU-Net: The ringed residual U-Net for image splicing forgery detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2019, pp. 1–10.

[45] K. Tomiyasu, "Tutorial review of synthetic-aperture radar (SAR) with applications to imaging of the ocean surface," *Proc. IEEE*, vol. 66, no. 5, pp. 563–583, May 1978.

[46] F. M. Henderson and A. J. Lewis, *Manual of Remote Sensing, Principles and Applications of Imaging Radar*, vol. 2, 3rd ed. Hoboken, NJ, USA: Wiley, 1998.

[47] C. Oliver and S. Quegan, *Understanding Synthetic Aperture Radar Images*. Rijeka, Croatia: SciTech, 2004.

[48] A. Moreira, P. Prats-Iraola, M. Younis, G. Krieger, I. Hajnsek, and K. P. Papathanassiou, "A tutorial on synthetic aperture radar," *IEEE Geosci. Remote Sens. Mag.*, vol. 1, no. 1, pp. 6–43, Mar. 2013.

[49] Copernicus Program. *Copernicus Open Access Hub*. Accessed: Jan. 2, 2022. [Online]. Available: https://scihub.copernicus.eu/

[50] F. De Zan and A. M. Guarnieri, "TOPSAR: Terrain observation by progressive scans," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 9, pp. 2352–2360, Sep. 2006.

[51] European Space Agency. (Jun. 2021). *Product Slicing*. Accessed: Jun. 28, 2021. [Online]. Available: https://sentinel.esa.int/web/sentinel/technical-guides/sentinel-1-sar/products-algorithms/product-slice-handling

[52] NGA. (Nov. 2020). *World Geodetic System 1984*. Accessed: Jan. 2, 2022. [Online]. Available: https://earth-info.nga.mil/GandG/update/index.php?dir=wgs84&action=ws84

[53] European Space Agency. (Jun. 2021). *Radar Course 2*. Accessed: Jun. 26, 2021. [Online]. Available: https://earth.esa.int/web/guest/missions/esa-operational-eo-missions/ers/instruments/sar/applications/radar-courses/course-2

[54] Copernicus Program. *Sentinel-1 Mission User Guide*. Accessed: Jan. 2, 2022. [Online]. Available: https://sentinel.esa.int/web/sentinel/user-guides/sentinel-1-sar

[55] M. Schmitt, L. H. Hughes, and X. X. Zhu, "The SEN1–2 dataset for deep learning in SAR-optical data fusion," *ISPRS Ann. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. IV-1, pp. 141–146, Sep. 2018.

[56] M. Schmitt, L. H. Hughes, C. Qiu, and X. X. Zhu, "SEN12MS—A curated dataset of georeferenced multi-spectral Sentinel-1/2 imagery for deep learning and data fusion," *ISPRS Ann. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. IV-2/W7, pp. 153–160, Sep. 2019.

[57] J. Lukas, J. Fridrich, and M. Goljan, "Digital camera identification from sensor pattern noise," *IEEE Trans. Inf. Forensics Security*, vol. 1, no. 2, pp. 205–214, Jun. 2006.

[58] N. N. Vo and J. Hays, "Localizing and orienting street views using overhead imagery," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 494–509.

[59] M. D. M. Hosseini and M. Kirchner, "Unsupervised image manipulation localization with non-binary label attribution," *IEEE Signal Process. Lett.*, vol. 26, no. 7, pp. 976–980, Jul. 2019.

[60] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Math. Statist. Probab.*, vol. 1, 1967, pp. 281–297.

[61] G. Mclachlan and D. Peel, *Finite Mixture Models*. Hoboken, NJ, USA: Wiley, 2000.

[62] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the *EM* algorithm," *J. Roy. Stat. Soc. B, Methodol.*, vol. 39, no. 1, pp. 1–22, 1977.

[63] J. Wei, Y. Zhang, H. Wu, and B. Cui, "An efficient change detection for large SAR images based on modified U-Net framework," *Can. J. Remote Sens.*, vol. 46, no. 3, pp. 272–294, May 2020.

[64] Z. Zhang, Q. Liu, and Y. Wang, "Road extraction by deep residual U-Net," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 5, pp. 749–753, May 2018.

[65] M. Alfarhan, M. Deriche, and A. Maalej, "Robust concurrent detection of salt domes and faults in seismic surveys using an improved UNet architecture," *IEEE Access*, early access, Dec. 10, 2020, doi: 10.1109/ACCESS.2020.3043973.

[66] N. Siddique, S. Paheding, C. P. Elkin, and V. Devabhaktuni, "U-Net and its variants for medical image segmentation: A review of theory and applications," *IEEE Access*, vol. 9, pp. 82031–82057, 2021.

[67] R. Deng, C. Shen, S. Liu, H. Wang, and X. Liu, "Learning to predict crisp boundaries," in *Computer Vision—ECCV 2018*. Berlin, Germany: Springer, 2018.

[68] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 318–327, Feb. 2020.

[69] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, Dec. 2014, pp. 1–15.

[70] P. Yakubovskiy. (2019). *Segmentation Models*. Accessed: Jan. 1, 2022. [Online]. Available: https://github.com/qubvel/segmentation_models

[71] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2019, pp. 6105–6114.

[72] N. Bonettini, E. D. Cannas, S. Mandelli, L. Bondi, P. Bestagini, and S. Tubaro, "Video face manipulation detection through ensemble of CNNs," in *Proc. Int. Conf. Pattern Recognit. (ICPR)*, Jan. 2021, pp. 5012–5019.

[73] S. Mandelli, N. Bonettini, P. Bestagini, and S. Tubaro, "Training CNNs in presence of JPEG compression: Multimedia forensics vs computer vision," in *Proc. IEEE Int. Workshop Inf. Forensics Secur. (WIFS)*, Dec. 2020, pp. 1–6.

[74] Y. Yousfi, J. Butora, J. Fridrich, and C. F. Tsang, "Improving EfficientNet for JPEG steganalysis," in *Proc. ACM Workshop Inf. Hiding Multimedia Secur.*, Jun. 2021, pp. 149–157.

[75] E. D. Cannas, S. Baireddy, E. R. Bartusiak, S. K. Yarlagadda, D. M. Montserrat, P. Bestagini, S. Tubaro, and E. J. Delp, "Open-set source attribution for panchromatic satellite imagery," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2021, pp. 3038–3042.

[76] C. M. Bishop, *Pattern Recognition and Machine Learning*. Berlin, Germany: Springer-Verlag, 2006

EDOARDO DANIELE CANNAS (Graduate Student Member, IEEE) received the M.Sc. degree in computer science and engineering from the Politecnico di Milano, Milan, in 2019, where he is currently pursuing the Ph.D. degree in information technology. Since 2019, he has been a Research Assistant within the Image and Sound Processing Laboratory, Politecnico di Milano. His research interest includes signal processing for multimedia forensics applications. In particular, he studies the development of forensic instruments for the remote sensing imaging field, and the forensic intepretability of machine learning and deep learning algorithms.

NICOLÒ BONETTINI (Graduate Student Member, IEEE) received the B.Sc. degree in computer science and engineering from the Università degli Studi di Modena e Reggio Emilia, in 2014, and the M.Sc. degree in computer science and engineering from the Politecnico di Milano, in 2017, where he is currently pursuing the Ph.D. degree with the Image and Sound Processing Laboratory. His research interests include signal processing applied to hyperspectral X-ray imaging and multimedia forensics.

SARA MANDELLI received the M.Sc. degree in telecommunications engineering and the Ph.D. degree in information technology from the Politecnico di Milano, Milan, Italy, in 2015 and 2020, respectively. She is currently a Postdoctoral Researcher with the Image and Sound Processing Laboratory, Politecnico di Milano. Her research interests include signal processing applied to images and videos in multimedia forensics.

PAOLO BESTAGINI (Member, IEEE) was born in Novara, Italy, in 1986. He received the M.Sc. degree in telecommunications engineering and the Ph.D. degree in information technology from the Politecnico di Milano, Italy, in 2010 and 2014, respectively. He is currently an Assistant Professor with the Image and Sound Processing Laboratory, Politecnico di Milano. His research interests include multimedia forensics and acoustic signal processing for microphone arrays. He has been elected as a member of the IEEE Information Forensics and Security Technical Committee and a Co-organizer of the IEEE Signal Processing Cup 2018. He is also an Associate Editor of the IEEE Transactions on Circuits and Systems for Video Technology.

STEFANO TUBARO (Senior Member, IEEE) was born in Novara, Italy, in 1957. He received the degree in electronic engineering from the Politecnico di Milano, Milan, Italy, in 1982. Then, he joined the Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, first as a Researcher with the National Research Council, and then as an Associate Professor, in November 1991. Since December 2004, he has been appointed as a Full Professor in telecommunication with the Politecnico di Milano. He is the author of more than 180 scientific publications on international journals and congresses and the coauthor of more than 15 patents. His current research interests include advanced algorithms for video and sound processing. In the past few years, he has focused his interest on the development of innovative techniques for image and video tampering detection and, in general, for the blind recovery of the processing history of multimedia objects. He coordinates the research activities of the Image and Sound Processing Laboratory, Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano. He had the role of a Project Coordinator of the European Project ORIGAMI (A new paradigm for high-quality mixing of real and virtual) and the Research Project ICT-FET-OPEN REWIND (REVerse engineering of audio-VIsual coNtent Data). This last project was aimed at synergistically combining principles of signal processing, machine learning, and information theory to answer relevant questions on the past history of such objects. He is a member of the IEEE Multimedia Signal Processing Technical Committee and the IEEE SPS Image Video and Multidimensional Signal Technical Committee. He was in the organization committee of a number of international conferences, including the IEEE MMSP 2004/2013, IEEE ICIP 2005, IEEE AVSS 2005/2009, IEEE ICDSC 2009, IEEE MMSP 2013, and IEEE ICME 2015. From May 2012 to April 2015, he was an Associate Editor of the IEEE Transactions on Image Processing. He is currently an Associate Editor of the IEEE Transactions on Information Forensics and Security.