IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 30 June 2025, accepted 15 August 2025, date of publication 22 August 2025, date of current version 28 August 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3601854

RESEARCH ARTICLE

Source Verification of Printed Logos for Anti-Counterfeiting Applications

NISCHAY PURNEKAR[1], (Student Member, IEEE), GIACOMO CANCELLI[2], ANSELMO FERREIRA[1], (Member, IEEE), AND MAURO BARNI[1], (Fellow, IEEE)

[1]Department of Information Engineering and Mathematics, University of Siena, 53100 Siena, Italy
[2]ViDiTrust S.r.l, 53100 Siena, Italy

Corresponding author: Nischay Purnekar (nischay.purnekar@student.unisi.it)

This work was supported in part by European Union Marie Skłodowska-Curie Project PrintOut under Grant 892757; and in part by Security and Rights in the Cyber Space (SERICS) Project under Ministero dell'Università e della Ricerca (MUR) National Recovery and Resilience Plan, funded by European Union (EU)—NextGenerationEU under Grant PE00000014.

ABSTRACT With the advent of the digital era, the growing prevalence of counterfeit goods poses serious threats to consumer safety and brand integrity, necessitating the development of novel techniques to combat forgery and counterfeiting. This paper focuses on the authentication of printed logos. The problem is formulated within a verification framework and is addressed using three distinct Siamese Neural Network (SNN) architectures: two shallow-and-wide networks and one based on the Xception model. Specifically, we investigate three SNN variants: Multiple Convolutions Summation (MCS-SNN), Multiple Convolutions Concatenation (MCC-SNN), and Mini-Xception SNN. To enhance authentication effectiveness, we adopt a two-step authentication process, beginning with an initial coarse analysis based on chromatic features, followed by a detailed verification that leverages micro-geometric, printer-specific artifacts, with each step focusing on distinct authentication patterns. The results obtained in both closed and open-set conditions indicate promising authentication accuracies, demonstrating the effectiveness of our approach for robust identity verification in real-world scenarios. Furthermore, the lightweight design of these models underscores their practical suitability for deployment on consumer-grade devices, highlighting their potential for real-world anticounterfeiting applications.

INDEX TERMS Authentication, counterfeit detection, geometric printer signatures, open-set verification, printed logos, Siamese neural network, two-step verification.

I. INTRODUCTION

The ever-growing economy characterizing today's globalized and interconnected world has significantly altered and improved the quality of people's lives, but it has also come with its share of problems. One of them is the widespread presence of counterfeited goods, which not only damages the reputation and financial stability of legitimate brand owners but also puts consumers' safety and confidence at risk.

As a matter of fact, the trade of counterfeited goods has evolved into a multi-billion-dollar industry. According to estimates of the World Health Organization [1], up to half of the malaria drugs in low and middle-income countries may be counterfeited. Even before the COVID-19 pandemic accelerated, the illicit trade in counterfeited and pirated goods had already reached significant proportions. One might assume that extensive exposure to certain brands through various media platforms would enable consumers to accurately identify them. However, research suggests that despite high exposure, consumers frequently make mistakes in differentiating between fake and genuine products—even those of well-known brands [2].

In this paper, we propose a system for the authentication of goods through printed logos. Logos play a major role in linking a product to its brand, as every brand is associated with a logo that serves as a symbol of its identity. Specifically,

The associate editor coordinating the review of this manuscript and approving it for publication was Md. Moinul Hossain.

we use printed logo authentication as a proxy for verifying the authenticity of the goods associated with the logo (*e.g.*, through a label). While logo authentication cannot fully replace product authentication, it offers the advantage of not requiring expensive and time-consuming mechanical or chemical analyses. As such, it can serve as a first step in combating the proliferation of counterfeited products.

The most common approaches for printed-logo and label authentication employ sophisticated anti-counterfeiting technologies, such as the use of special inks [3], watermarking [4], and laser holography [5], among others. These technologies involve creating anti-counterfeiting labels using specific materials or production processes. While effective, such systems are often costly, as they require modifications to the label generation process and/or the use of special and expensive materials.

The anti-counterfeiting system discussed in this paper adopts a simpler approach, relying on the verification that the logo was printed using an authorized printer. In principle, we do not make any assumptions about the specific printer used to print the logo,[1] nor do we require any modifications to the printing process in any way. Printer identification is based on the analysis of unique chromatic artifacts and micro-geometric patterns introduced by the printer. Their presence serves as tangible proof of the brand's identity and, by extension, of the authenticity of the goods. Figure 1 provides a visual representation of the proposed label authentication pipeline.

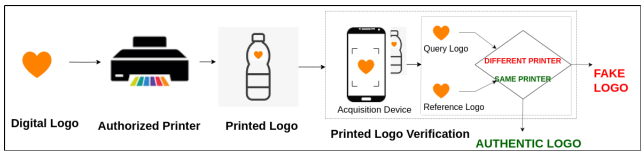


FIGURE 1. Overview of the proposed printed logo verification pipeline. The system verifies a printed logo's authenticity by comparing printer-specific characteristics of a query logo, captured via a mobile device, with those of a reference logo printed by an authorized printer. A match indicates authenticity; otherwise, the logo is flagged as fake.

A challenge we have to face is to ensure the effectiveness and reliability of the authentication process in the presence of low-quality, general-purpose acquisition devices, and unpredictable environments with non-controlled acquisition conditions, that is, devices that are not specialized for a specific authentication service but are instead intended for general use. This includes smartphones, which are ubiquitous and widely available, but may not always have the highest quality components or specifications. Non-controlled acquisition conditions refer to situations where the environment in which the system operates is not completely controlled or predictable. This could include situations where the lighting is poor, the device is not held steadily, or the device is in motion. An additional challenge tackled by our system is that the authentication system should be able to operate in an open-set scenario, where the to-be-authenticated logo and the printer used to print it are not necessarily included in the datasets used during training. In this way, we avoid having to retrain the system whenever a new logo or a new printer must be authenticated.

As shown in Figure 1, we tackle these issues by presenting a Machine Learning verification system capable of verifying if the query logo is a genuine or counterfeited logo, by comparing it with a reference logo printed by an authorized printer through Deep Learning (DL) networks. In this way, it is only necessary that the Deep Neural Network (DNN) be able to recognize if two printed logos have been printed by the same printer or not, regardless of whether the printer and the logos have been seen before or not. To simplify the analysis, we focus on uniform color logos; however, the system may be easily adapted to work with non-uniform logos.[2] More specifically, we employ several shallow-and-wide Convolutional Neural Networks (CNNs) in a Siamese Neural Network (SNN) setting. Such networks learn image similarities and dissimilarities between image pairs to distinguish if the two images given at the input of their two branches have been printed by the same printer or not. In this way, authentication is achieved by comparing an unknown logo with an authentic logo printed by an authorized printer.

A unique feature of our proposed approach is that verification is performed through a two-step authentication procedure, consisting of the application of two cascaded SNNs. The first SNN is in charge of distinguishing authentic logos from, so to speak, *easy* fakes, which are fake logos whose appearance (*e.g.*, their overall color) is quite different from that of authentic logos. This first model is trained by showing the SNN both matching and non-matching logo pairs.[3] The non-matching pairs, hereafter referred to as fake pairs, have been gathered by considering a wide variety of non-matching logos with different colors and printer artifacts. As a result, the first SNN tends to rely on rather evident chromatic differences and does a good job of spotting the easy fakes, but it fails to recognize high-quality, hard-to-detect fakes (logos with colors similar to the authentic ones). For this reason, the pairs that pass the first authentication step are further evaluated by a second SNN model, which has been fine-tuned to discriminate hard pairs by relying upon the tiny geometric artifacts introduced by different printers.

An additional problem we have to face is the impossibility of building large training datasets, due to the inherent difficulties associated with data acquisition through mobile devices. For this reason, our system is based on shallow-and-wide architectures that can work efficiently even with

[1] In this paper, we focus on professional printers, namely offset and flexographic printers.

[2] Indeed, the authentication of non-uniform logos may be even easier, since in this case, we expect a larger variability between the patterns introduced by different printers.

[3] A matching pair for the first model is a pair of logos of the same color printed by the same printer, while in a non-matching pair, the two logos have different colors and are also very likely been printed by different printers.

a limited amount of training data. The experiments we ran showcase promising results even when confronted with unknown logos that were not included in the training set, ensuring reliability in open-set conditions. Our comprehensive evaluation, carried out on a challenging, realistic dataset under closed-set and open-set conditions, highlights the commendable performance of the proposed solution and its potential for real-world applications.

Given the above discussion, the contributions of our paper can be summarized as follows:

1) We propose a printed logo authentication procedure based on Siamese Networks to authenticate counterfeited goods, treating such an issue as a printed logo verification problem. The Siamese networks compare any new sample logo to a set of reference (or gallery) of known authentic samples, giving their verdict regarding authenticity.
2) We propose a new printed logo authentication system based on a two-step verification procedure, relying on a cascade of two SNNs, in charge of detecting fake logos based, respectively, on chromatic (color) and geometric (printer) artifacts.
3) In real-world scenarios, deep CNNs demand substantial data for effective training. To tackle this challenge, we propose using shallow-and-wide architectures. These specialized models can be trained from scratch on small datasets (dozens of thousands of samples). The skeleton of our networks is derived from [6], with hyperparameters tweaked to better address the problem at hand.
4) We demonstrate the reliability and learning capabilities of our method under open-set conditions. A task that is important to assess the functionality of the system in a more real world scenarios.

The rest of the paper is organized as follows. In Section II, we summarize related work on printed document authentication. In Section III, we present the settings, architectures, and training schemes of our authentication system. Section IV presents the dataset and experimental setup used to validate our method. Section V shows and discusses experimental results under both closed- and open-set conditions. Finally, Section VI concludes the paper by pointing out future research directions.

II. RELATED WORK

In the anti-counterfeiting framework, several works have proposed attaching specific anti-counterfeiting labels to documents and goods and analyzing only such labels to detect counterfeiting. One of the first works in this direction relied on a chemical analysis of the printer ink [7], [8]. Digital image processing-based solutions for anti-counterfeiting purposes have also evolved over the last decade, motivated by the fact that chemical and physical procedures on hard copy documents can damage the document and require expensive staff and machinery. Such approaches exploit printing signatures known as extrinsic (included in the document, detected by active forensic procedures) and intrinsic (natural to each printer, detected by passive forensic methods) signatures [9].

Methods based on extrinsic signatures include an external signature or label in every printed document, making it impossible to faithfully duplicate the document. By employing particular printing techniques and materials that are not available to counterfeiters, illegal copy attacks on printed materials can be successfully avoided.

One of the first approaches in document authentication with extrinsic signatures relies on so-called Copy Detection Patterns (CDPs), in which a label could be printed by any printer, and the reliability of the label relies on who possesses the label and prints it. Among other efforts in this direction, the work from [10], [11] introduced a two-level QR code (2LQR code) to be attached to any document and used for authentication. Due to the uncertainty involved in the engraving process to carry out authentication, [12] demonstrated that each engraved cylinder in rotogravure printing should also be considered to perform anticounterfeiting analysis. Reference [13] combined visual features with QR codes to design an Internet of Things anti-counterfeiting system. The visual features encompassed natural texture features and printed micro-features. During the anti-counterfeiting verification process, a feature extraction algorithm computes the similarity between the test sample and the database sample, yielding identification results. Reference [14] proposed a two-step multilevel barcode halftone descriptors method to discriminate pristine multilevel barcodes and their counterfeited counterparts. Reference [15] authenticated the same barcode in a feature fusion approach, using halftone local and global descriptors. Reference [16] introduced the LCAC (Low-Cost Anti-Copying) 2D barcode, which achieves anti-duplication by incorporating confidential authentication information into the original data during QR code generation. Reference [17] proposed an authentication scheme for anti-counterfeiting patterns captured by smartphones. Their approach employs a single classifier with two modules: a U-Net-based feature extraction module and a boundary-optimized One-Class Support Vector Machines (SVMs) classification method. Experimental results demonstrate the method's superior ability to distinguish genuine and counterfeited pattern images, achieving 100% precision and recall rate and significant enhancement in the recognition of blurry images of genuine anti-counterfeiting patterns. Reference [18] proposed a system for QR code verification using various CNN architectures. Reference [19] suggested an anti-counterfeiting solution using a binary anti-counterfeiting pattern with small random textures. Some other works based on extrinsic signatures include [4], which proposes a technique based on steganography where the verification information is hidden within the document itself, and [20] proposed a technique that uses watermarking to embed a unique identifier into the printed document. Techniques based

on extrinsic signatures have some drawbacks, including the fact that they are expensive, require specific materials and printing procedures, not all printer brands support them, and some of these signatures can be masked in printed materials [21].

In contrast to extrinsic signatures, which are content-dependent and specific to a document, intrinsic signatures are inherent to the printing device [22]. Due to imperfections in the electromechanical component of the devices, these signatures are unintentionally embedded in the printed document and can be used to detect their source. Such signatures can be found by scanning a printed document at a high resolution and using image analysis to extract features invisible to the naked eye. Banding, jitter, and skewed jitters are some of the most popular intrinsic signatures [23]. Most of the works in this class engineer such signatures into features for a multi-class (or even one-class) printer source attribution classification scenario. The literature has mostly concentrated on such signatures, which are also the focus of the solution put forward in this paper.

To identify the source printer through intrinsic signatures, the approaches for text (non-colored) documents often use handmade features based on printing imperfections. These features are taken from a small subset of the data (such as one symbol or letter) and are input to supervised classifiers. For instance, in one of the pioneering studies in this area, [24] retrieved the letter "I" from the printed text using raw pixel values in a multi-class machine learning system to distinguish various source printers. The source printer of an unknown document is determined by majority voting on each letter "I". Such a pipeline of image descriptions served as a model for other works. Some examples are Gray Level Co-occurrence Matrices (GLCMs) features from [25], [26], [27], and [28] and [29], *ad-hoc* texture descriptors from [30] and [31], SURF and ORB features from [32], and geometric distortions signatures from [33].

In the case of non-textual printed data, which is the subject of this research, intrinsic artifacts are more frequently discovered as more sections of the picture are printed. Reference [9] used the Fourier transform of retrieved image patches to distinguish between distinct banding frequencies in color laser printers. Reference [34] presented a novel colored documents dataset, containing recent industrial and office laser printers, and evaluated the capability of several CNNs to identify the source devices. Reference [35] suggested a DL technique to extract features from specific printed microscopic patterns, feeding such features into multi-class SVMs and a Random Forest classifier for source identification. Reference [13] employed natural texture features and printing micro-features to determine the feature similarity between the QR code being tested and the sample QR code using a feature extraction algorithm. However, this approach necessitates a high-precision printing and capturing environment. In prior research by [36], efforts have been made to enhance QR code security, exemplified by the development of texture-hidden QR codes to mitigate counterfeiting risks. However, challenges persist in effectively recovering and authenticating codes under adverse conditions, necessitating continued investigation, particularly in light of emerging forgery technologies. Other solutions for source attribution of color documents include the analysis of geometric distortions [37], [38] and halftone texture descriptors [39], [40], [41].

The work in [42] introduced a Bottleneck Residual Block (BRB) and used it to design a CNN-based architecture termed PSINet. They evaluated the impact of parameters such as input image size, convolution kernel size, and the number of convolutional layers on the performance of the proposed model for printer source identification. Building on these insights, the authors presented an enhanced CNN model, SE-BRB-Net, which integrates the BRB with a squeeze-excitation attention mechanism [43]. The improvements enabled the model to better emphasize printer-specific features while suppressing irrelevant information. Both proposed modules demonstrated strong performance for printer identification. References [44] and [45] also developed customized CNN models tailored for analyzing both grayscale and color image data in the context of printer source identification. Reference [46] investigated differences between original and compatible toner cartridges to predict characteristics of the source printing device. Their approach involved extracting 20 texture-based features derived from image quality metrics, GLCM, Segmentation-based Fractal Texture Analysis (SFTA), and LBP, using microscopic white-light and laser confocal microscopy images. The classification was performed using a combination of Mixed Discriminant Analysis (MDA) and Flexible Discriminant Analysis (FDA), effectively predicting the manufacturer, model, and cartridge type. Reference [47] employed a ResNet50-based model to classify printers using images transformed via Fast Fourier transform (FFT), which maps spatial-domain input into the frequency domain to enhance training efficiency. The classification was carried out using the central regions of the FFT images from the cyan color channel. Reference [48] proposed a StarDist-based method that integrates object detection and segmentation to distinguish between laser printers of the same brand. Their analysis revealed distinguishable characteristics among printing samples. In a related study, [49] applied a watershed segmentation method that combines geomorphology- and region-based techniques, followed by CNN-based classification for printer identification. A final category of intrinsic signature techniques includes those that can handle any type of document, including text documents and color images. For instance, [29] described an image descriptor strategy based on GLCMs in conjunction with a Convolutional Texture Filter approach to be used on image frames, which are regions of the printed page with enough printed material. Reference [50] looked into the printer attribution using shape descriptors with support vector machines and random forest

classifiers. Finally, it was demonstrated that when enough data is available for training, DNNs are able to directly learn the features from the data [51], [52], [53].

Although the use of signatures attached to documents printed with specific techniques (such as CDPs) or printing materials (such as extrinsic signatures) has high security, they raise the cost, complexity, training time, and capacity of the datasets used for the training and testing. As a result, printed logos and digital anti-counterfeiting techniques are receiving continuous interest from academia and industry as they are simpler to produce. We thus focus on this kind of authentication throughout the rest of this paper and detail how to use logos in our authentication procedure in the next section. An additional advantage of the method proposed in this paper with respect to the state of the art, besides the fact that it is applied to authenticate logos (which are cheaper and simpler than CDPs and other extrinsic signature methods), is its use of shallow SNNs, which require less data to be trained than common DNN solutions used for intrinsic signatures authentication.

III. METHODOLOGY

The system proposed in this paper addresses some limitations of existing state-of-the-art approaches: the first one is the inability to work in an open-set scenario (for example, works on source attribution usually consider a limited class of known printers). The second one is the lack of a second layer of authentication, where high-quality forgeries can be treated separately (which is not properly explored in the CDP authentication literature, for example). The approach proposed in this paper treats these limitations by verifying whether a logo attached to a commodity is genuine or counterfeited, by comparing it with a reference logo (or template) printed by an authorized printer, and, hence, can easily be adapted to work in an open-set scenario. Our approach involves a two-step verification procedure to address the authentication of printed logos with shallow Siamese networks. By authenticating logos with a two-step procedure, we verify the logos' authenticity by considering different levels of forgeries, making the proposed method difficult to attack. Additionally, by focusing on logos with constant color patches, the authentication process is streamlined as they are easy and cost-effective to create. We detail the proposed method in the next subsections.

A. SIAMESE NEURAL NETWORKS

Siamese Neural Networks were initially introduced in the early 1990s by [54] as a solution for signature verification, treating it as an image-matching problem. A SNN consists of two twin networks, as illustrated in Figure 2. Such networks accept separate inputs but are interconnected by a distance loss function. This function calculates a distance metric between the high-level feature representations extracted by each branch. The parameters of the twin networks are identical, ensuring that two highly similar images are not mapped to different locations in the feature space. Additionally, the network architecture is symmetric, so the order in which the two input images are presented to the twin networks is irrelevant, as the top layers compute the same metric even if the images are given to the opposite twins. Eventually, the outputs of the twin networks are compared by using a similarity metric to determine the similarity or dissimilarity between the input pairs.

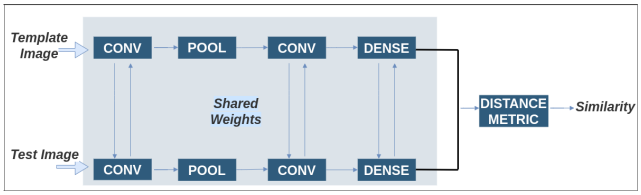


FIGURE 2. Architecture of the Siamese network for image similarity verification. Two identical networks with shared weights process the template and test images in parallel, and their outputs are compared using a distance metric to compute a similarity score.

The distance used to compute the dissimilarity between the images is usually the Euclidean distance. Given two points $(x_{11}, x_{12}, x_{13},\ldots x_{1n})$ and $(x_{21}, x_{22}, x_{23},\ldots x_{2n})$ in the feature space, we simply have:

$$d = \sqrt{\sum_{i=1}^{n} (x_{1i} - x_{2i})^2}. \tag{1}$$

To minimize the above distance function for pairs with similar images and maximize it for pairs of dissimilar images, the two networks share a unique loss function. In our case, the networks were trained by using a binary cross-entropy loss function, defined as:

$$L(y, \hat{y}) = -\frac{1}{N} \sum_{i=1}^{N} y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i), \tag{2}$$

where y_i is the true binary label (0 or 1), $\hat{y}_i$ is the predicted similarity score, and N is the number of pairs.

Siamese networks also facilitate few-shot learning, which is effective training with few samples. This happens because, by training with pairs instead of individual images, there is an ability to significantly expand the training dataset for limited sample scenarios. For instance, if the number of classes is denoted by M, and the sample count per class is N, we have a total amount of data equal to $M \times N$. By adhering to the Siamese network's structure, the samples are paired with each other, resulting in a total number of training pairs given by

$$Y_{pair} = \frac{(M \times N)!}{2!(M \times N - 2)!} \tag{3}$$

In summary, SNNs have the following advantages over common CNNs: (i) SNNs typically require less training data compared to other DL models. This makes them useful in scenarios where labeled data is scarce or expensive to obtain; (ii) SNNs can be used with a wide variety of data types, including text, images, speech, and other modalities. This makes them a versatile solution for a wide range of

applications; (iii) SNNs can provide similarity scores or distances between pairs of data points, making them useful for tasks where explainability is important; and (iv) SNNs can be easily scaled to handle large datasets and high-dimensional data. The parallel architecture of Siamese networks also allows efficient training on GPUs, making them suitable for large-scale applications.

B. AUTHENTICATION PIPELINE

Before delving into the details of our authentication method, we introduce two concepts that will be used in the rest of the paper: the subdivision of the logo into blocks and majority voting. Our logos are regular patterns with 192 × 192×3 resolution. These patterns are subdivided into 9 image blocks with 64 × 64×3 resolution. Such blocks will be the data verified by the SNN authenticator. Our authentication method uses one round of majority voting to define a logo as authentic or fake. Such a round uses the majority voting of the above-mentioned 9 blocks classification performed by SNNs to define each logo's authenticity. The majority voting step is particularly efficient when part of the logos are difficult to classify (due to non-constant illumination distribution over different blocks). Figure 3 shows the overall pipeline of the proposed verification system. In the first step, all the blocks composing the to-be-authenticated logo are analyzed independently, by comparing them against authentic logo blocks present in an authentication-only dataset kept separately. The SNN in the first authentication step relies mainly on the chromatic content of the blocks, roughly detecting if the pair of logos has similar colors or not. Chromatic artifacts stem from differences in ink distribution and color calibration between printers. These differences manifest in the overall hue, saturation, and consistency of the printed colors, especially in uniformly colored regions like those found in our logo dataset. The first SNN is trained to pick up on these macroscopic color-level differences, making it effective at identifying obvious or low-quality fakes where the ink or color tone deviates from the authentic samples. This makes the first SNN vulnerable to illegal copy attacks, where the counterfeiter simply scans and reprints an authentic logo with another printer trying to imitate the chromatic content of the authentic logo. Therefore, in the subsequent stage, a second SNN fine-tuned on hard fakes is applied only to block pairs that have passed the first authentication step. The second SNN targets geometric artifacts such as micro-patterns that are imperceptibly embedded during the printing process due to mechanical and hardware imperfections specific to each printer. These include banding effects, microscopic dot patterns, and subtle alignment variations that are not reproducible by counterfeiters. By training the second network on such hard examples, we enable the model to focus on these high-frequency, fine-grained textural and structural cues that go beyond color fidelity. This makes it effective against high-quality forgeries that may have chromatically matched the original but lack the correct geometric signature. Eventually, a pair of blocks is considered authentic only if they pass the analysis of the two SNNs. The final decision on the authenticity of the whole logo is made by applying majority voting to the results obtained at the block level.

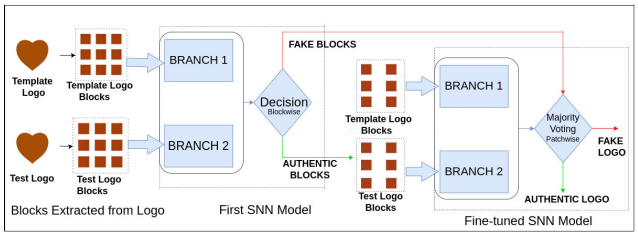


FIGURE 3. Proposed logo verification pipeline using a Siamese Neural Network (SNN). The template and test logos are divided into blocks, which are compared pairwise using the first SNN. Blocks predicted as authentic are further analyzed by a fine-tuned SNN, and final authenticity is determined through majority voting.

Together with the trained SNN models, we also need a database of authentic templates to be used within the two-step authentication system. When the system is asked to work on new logos, the reference examples for the new logo must be added to the database of authentic templates; however, the SNNs do not need to be retrained.

In the next section, we describe in more detail the architectures that we have used to build the SNNs.

C. SIAMESE NETWORKS ARCHITECTURES

The choice of the SNN architecture is dictated by the difficulty of gathering large amounts of training data. Building upon previous research on steganalysis [55], [56], we employed a tweaked version of the shallow-and-wide SNN architectures proposed by [6]. We also tested a shallow version of the Xception network in a Siamese setting.

1) MULTIPLE CONVOLUTIONS SUMMATION SIAMESE NEURAL NETWORK (MCS−SNN)

The initial architecture of our proposed method includes a Multiple Convolution Summation Siamese Neural Network (MCS−SNN) model, inspired by RESNET [57], but following a distinct approach. Figure 4 shows the pipeline of MCS−SNN.

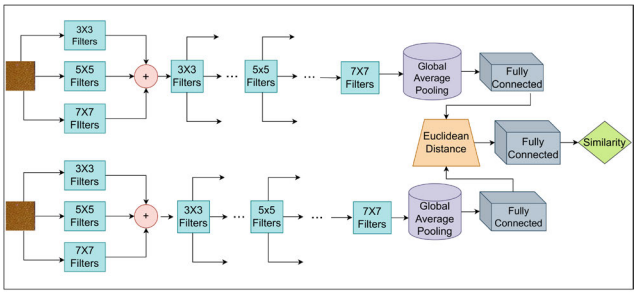


FIGURE 4. MCS-SNN architecture. Multiscale features are extracted using parallel convolutional branches and fused via element-wise summation. The output is processed through convolutional layers, global average pooling, and a fully connected layer. Similarity is computed via Euclidean distance and refined by an additional fully connected layer.

Instead of deep layers, MCS-SNN employs multiple convolutional modules in parallel, generating multiple feature maps that are then fused and directed through the network. This process is repeated three times, resulting in a shallow yet wide model. All of these 3 sets of layers in parallel consist of 32 filters with 3×3, 5×5, and 7×7 filter sizes, each of them followed by max pooling, batch normalization, and 30% dropout. After such parallel convolutions are summed, there is a pooling operation followed by a dropout of 30%. Then, 3 sets of individual single convolutional layers with 32 filters with 3×3 or 5×5 or 7×7 dimensions receive and process the summations of previous parallel layers, with LeakyRELU [58] activation applied later. The model at the end creates pairs of 48-dimensional vectors through global average pooling and fully connected layers, which are compared using the Euclidean distance after passing through a sigmoid-function layer.

2) MULTIPLE CONVOLUTIONS CONCATENATION SIAMESE NEURAL NETWORK (MCC-SNN)

The second shallow-and-wide SNN is based on the [59] Inception modules. In such a module, the feature maps resulting from the parallel convolutional layers are concatenated. These SNNs are similar to the previous MCS-SNN, with the exception that they concatenate the feature maps rather than summing them. More details can be seen in Figure 5.

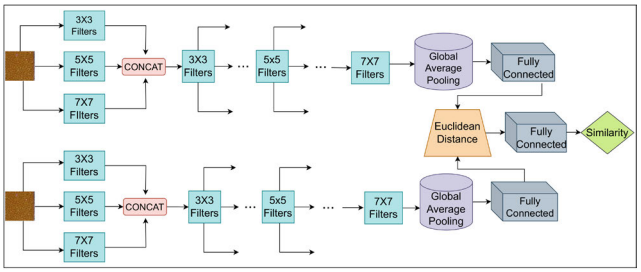


FIGURE 5. MCC-SNN architecture. Similar to MCS-SNN, but multiscale features are combined via channel-wise concatenation. The network follows with convolutional layers, global average pooling, and fully connected layers to compute and refine the Euclidean distance-based similarity score.

3) MINI-XCEPTION SIAMESE NEURAL NETWORK

The Xception design [60] inspired the final set of SNNs used in our experiments. In DL frameworks, a depthwise separable convolution consists of a spatial convolution, performed independently over each channel of an input, followed by a pointwise convolution (*i.e.* a 1×1 convolution), projecting the channel's output by depthwise convolution onto a new channel space. In a modified separable depthwise convolution, an initial pointwise convolution is performed to modify the dimensions of the image, followed by depthwise convolutions.

The architecture of the Xception network is made up of three flows: entry, middle, and exit flow, with the middle flow repeated several times throughout the network. However, in our Mini-Xception case, we use only one middle flow, which differs from the original Xception architecture, making it shallower. The details of the SNN pipeline can be seen in Figure 6. The overall architecture makes use of ReLU activation. Following all the convolution procedures, the features are converted again to 48-dimensional vectors that show how similar or dissimilar two image patterns are in the exit flow.

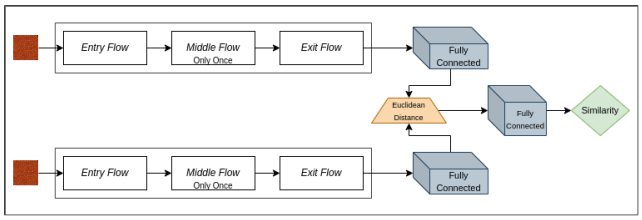


FIGURE 6. Mini-Xception SNN architecture. Each input passes through a lightweight Xception module with entry, middle, and exit flows. Feature embeddings are compared via Euclidean distance and refined using a fully connected layer.

IV. EXPERIMENTAL SETUP

In this section, we describe the datasets used for training, fine-tuning, and testing purposes, plus the methodology we used to train the networks forming the two-step authentication system.

A. DATASET

The dataset used for logo authentication comprises five types of logos printed by authorized printers with different colors and a large, diverse fake dataset with logos printed by non-authorized printers, as can be seen in Table 1. These logos are uniformly distributed in colors, and each color of the logo has a printed pattern with unique artifacts generated in the printing process. All the logos were printed by offset and flexographic printers. The printed logos have been scanned using commercial mobile phones. A non-exhaustive list of the phones used is provided in Table 2.

The colors of the authentic logos are Dark Orange, Light Orange, Dark Blue, Light Blue, and Green, while the fake logos have different colors and patterns. The dataset of fake logos includes several colors (including unknown logo colors and colors similar to those of the authentic logos). The fake logos were obtained by printing them with unauthorized printers. The fake samples include both easy and hard fakes. Such a dataset is used to evaluate the performance of the SNNs in both closed-set and open-set settings.

To train the Initial model, we separated some of the Dark Orange, Light Orange, Dark Blue, and Light Blue logos, in addition to some fake logos. We left out the Green logo for the open-set experiments reported later in the paper.

The dataset used in this study is not publicly available due to proprietary constraints. However, it can be shared for

TABLE 1. Composition of the Printed Logo Dataset. The dataset includes logo patches from both authorized and unauthorized printers. Authentic logos span five color categories, while fake logos are produced by unauthorized printers. The table shows the number of patches per class with representative samples.

Dataset - Printed Logos		
Logo Type	*#Logo Patches*	*Examples of Logo Patches*
Logos Printed by Authorized Printers		
Dark Orange (D.O.)	8,320	
Light Orange (L.O.)	11,040	
Light Blue (L.B.)	7,280	
Dark Blue (D.B.)	1,016	
Green	7,536	
Logos Printed by Non-Authorized Printers		
Fake	347,648	

TABLE 2. Mobile phones Used for Logo Acquisition. A diverse set of smartphones from three major brands were used to capture printed logos under realistic conditions, introducing variability in resolution and imaging pipelines to improve system robustness and generalization.

Mobile Phones used	
Device Brand	*Models*
Samsung	S3, S4, S5, S6, S6Edge, S7, S7Edge, J5 2016, S10
Huawei	MATE S, Y5II, P9Plus, P9 LITE, P8 LITE
Apple	iPhone - 5c, 6, 7

research purposes upon signing a Non-Disclosure Agreement (NDA) with one of the co-authors.[4]

B. TRAINING METHODOLOGY

SNN training is carried out by initially subdividing the logos into pairs of authentic and fake logos for training, validation, and testing. The number of logos of the various colors is shown in Table 3. We obviously started generating splits at the logo level to guarantee that, later, blocks of the same logo are always spread over the same data split.

Later, each logo is further subdivided into 9 blocks of size 64 × 64 × 3. The blocks were then paired and used to train the networks. As the two SNNs of the two-step authentication scheme work differently, we had to employ a different number of logos for training and validation of each of them. The number of block image pairs per logo color used to train the first model of the two-step authentication chain is shown in Table 4. Figure 7 shows some examples of authentic and fake pairs for the first model. Fake pairs include pairs with marked chromatic differences between the blocks.

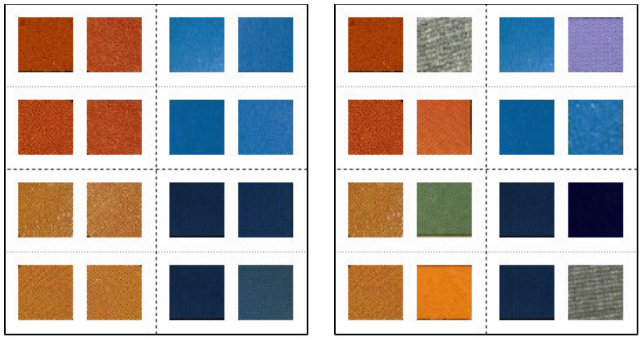
[4]Interested researchers may contact Giacomo Cancelli at giacomo.cancelli@gmail.com to request dataset access.

TABLE 3. Dataset Split – Train, Validation, and Test Sets. For each configuration, the number of authentic and fake logo images is reported. For instance, the (D.O, Fakes) entry includes 2062 original Dark Orange logos and 2062 fake logos (any color) used for training. These logos are later subdivided into (64 × 64 × 3) patches to generate input data for the networks.

Dataset Split First model			
Logo Type	*# Images (Logos)*		
	Train	*Validation*	*Test*
(D.O, Fakes)	(2062, 2062)	(1016, 1016)	(5242, 5242)
(L.O, Fakes)	(2070, 2071)	(1021, 1020)	(7949, 7949)
(L.B, Fakes)	(2048, 2049)	(1009, 1009)	(4223, 4222)
(D.B, Fakes)	(497, 498)	(214, 213)	(305, 305)

TABLE 4. Train and validation split of block pairs for the first model. The table reports the number of (64 × 64 × 3) block pairs used to train and validate the first Siamese network. Each pair includes a genuine and a fake logo block, organized by color type to ensure diversity and class balance.

Dataset Split first model		
Logo Type	*Image Pairs (# of Block pairs)*	
	Train	*Validation*
(D.O, Fakes)	(18558, 18558)	(9144, 9144)
(L.O, Fakes)	(18630, 18639)	(9189, 9180)
(L.B, Fakes)	(18432, 18441)	(9081 & 9081)
D.B & Fakes	(4473, 4482)	(1926, 1917)

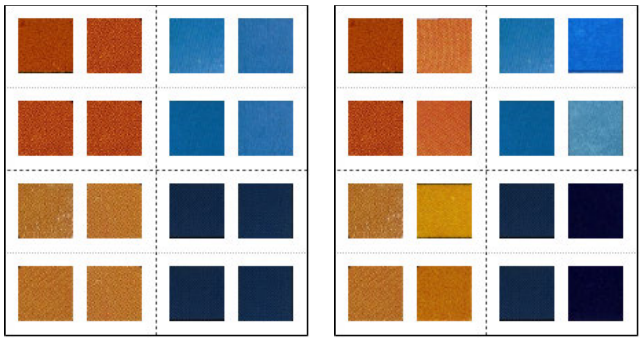


a Authentic image pairs b Fake image pairs

FIGURE 7. Examples of authentic and fake block pairs used in first-stage SNN training. Each pair is a (64 × 64 × 3) block from printed logos. (a) shows *authentic* pairs from the same authorized printer; (b) shows *fake* pairs combining blocks from authorized and non-authorized printers. These samples train the SNN to capture printer-specific differences for initial authentication.

The second SNN is built by fine-tuning the first network weights on hard fake pairs (*i.e.*, pairs of blocks of very similar colors but printed by different devices). The number of logos and blocks we used to fine-tune the second model and the way the dataset was split into training and validation subsets are shown in Tables 5 and 6, respectively. The reader may notice that, as we normally have fewer samples of hard fakes than easy fakes, the second model is trained with less data than the first one. In Figure 8, we show some examples of blocks used to train the second model. In this case, the fake pairs are much more difficult to distinguish, thus highlighting the necessity of a second model capable of handling these pairs.

As for the training hyper-parameters, for MCS−SNN we used the ADADELTA method from [61] in the first



a Authentic image pairs b Hard Fake image pairs

FIGURE 8. Examples of authentic and hard fake block pairs used in second-stage SNN training. (a) shows *authentic* pairs from the same authorized printer, while (b) shows *hard fake* pairs that closely resemble genuine ones. This stage focuses on challenging cases to improve the model's robustness against high-quality forgeries.

TABLE 5. Train and Validation Split of Logo Images for the Second SNN. The table shows the number of logo images used to train and validate the second Siamese model, which targets high-quality fake logos ("difficult fakes"). Each logo type includes a balanced number of samples to support robust learning and generalization.

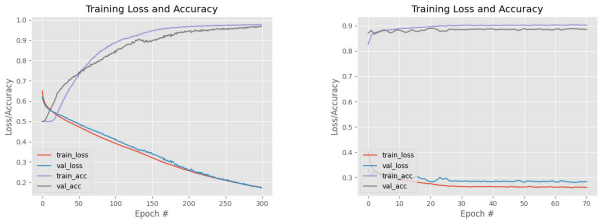
Dataset Split Second model		
Logo Type	*# Images (Logos)*	
	Train	*Validation*
(D.O, Difficult Fakes)	(1237, 1237)	(825, 825)
(L.O, Difficult Fakes)	(599, 599)	(323, 323)
(L.B, Difficult Fakes)	(247, 248)	(83, 82)
(D.B, Difficult Fakes)	(347, 348)	(150, 149)

TABLE 6. Train and Validation Split of Block Pairs for the Second SNN. The table reports the number of (64 × 64 × 3) block pairs used to train and validate the second Siamese model. Each pair combines authentic and difficult fake blocks to help the model detect subtle differences in high-quality forgeries.

Dataset Split Second model		
Logo Type	*Image Pairs (# of Block pairs)*	
	Train	*Validation*
(D.O, Difficult Fakes)	(11133, 11133)	(7425, 7425)
(L.O, Difficult Fakes)	(5391, 5391)	(2907, 2907)
(L.B, Difficult Fakes)	(2223, 2232)	(747, 738)
(D.B, Difficult Fakes)	(3123, 3132)	(1350, 1341)

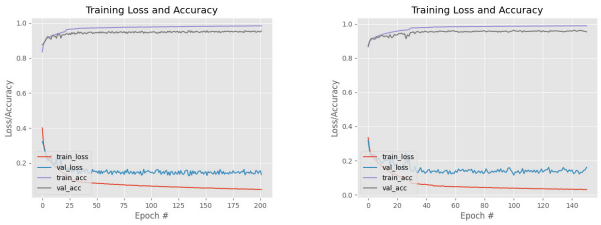
authentication step, and the AdaGRAD method from [62] for the second step SNN. For `MCC-SNN`, we used again the ADADELTA optimization for the first SNN, and NADAM optimization from [63] for the second network. Finally, for `Mini-Xception`, we used the RMSPROP optimizer from Geoffrey Hinton for the first step and the NADAM optimizer [63] for the second one. We used an initial learning rate of 0.001. Our batch size is 64 pairs of training blocks. The networks were optimized for 2,000 epochs, using early stopping, to stop training after the best accuracy had been obtained on the validation data, with a patience of 20 epochs before deciding to stop training. The Loss used is the Binary Cross Entropy, defined as in Equation (2).

In Figures 9, 10, and 11, we show the train and validation curves for the three SNNs architectures we have tested, for both the first and second authentication steps.



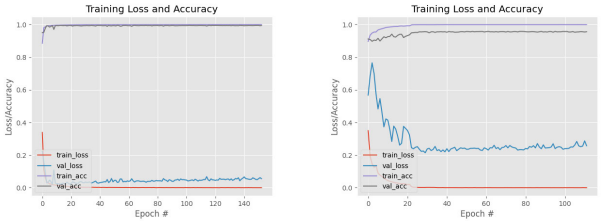
a First step `MCS-SNN` model. b Second step `MCS-SNN` model.

FIGURE 9. Training and validation curves (loss and accuracy) for both authentication steps using the `MCS-SNN` model. Red and blue curves represent training and validation loss, while purple and gray curves represent training and validation accuracy.



a First step `MCC-SNN` model. b Second step `MCC-SNN` model.

FIGURE 10. Training and validation curves for the `MCC-SNN` model. Color scheme: red (training loss), blue (validation loss), purple (training accuracy), and gray (validation accuracy).



a First step `Mini-Xception` model. b Second step `Mini-Xception` model.

FIGURE 11. Training and validation curves for the `Mini-Xception` model. Red and blue denote training and validation loss; purple and gray indicate training and validation accuracy.

C. METRICS

In this section, we introduce the metrics employed to evaluate the effectiveness of our approach. To clarify the metrics presented in the subsequent subsections, as well as the discussion throughout the rest of this paper, we define a *positive* sample as an authentic logo pair printed by the same printer, while a *negative* sample refers to counterfeited logos produced by different printers.

1) CONFUSION MATRIX

This matrix provides a detailed breakdown of the model's predictions by categorizing them into four outcomes: True Positives, False Positives, True Negatives, and False Negatives.

- True Negatives (*TN*): These are the cases where the model accurately classifies a pair of logos as *negative* (logos printed by different printers).
- False Positives (*FP*): These refer to instances where the model incorrectly predicts *positive* (authentic) a pair of *negative* logos (i.e., logos printed by different printers).
- True Positives (*TP*): These are the cases where the model correctly identifies a pair of logos as *positive* (i.e., authentic logos printed by the same printer).
- False Negatives (*FN*): These occur when the model mistakenly classifies an authentic pair of logos as *negative* (i.e., failing to recognize that the logos are printed by the same printer).

Table 7 summarizes the structure of the confusion matrix used for evaluating our binary classification performance. This matrix is instrumental in computing standard performance metrics such as accuracy, precision, recall, and F1-score.

TABLE 7. Structure of the confusion matrix used in our logo authentication system, indicating how predictions are categorized into True Negatives (*TN*), False Positives (*FP*), False Negatives (*FN*), and True Positives (*TP*).

	Predicted Negative	Predicted Positive
Actual Negative	*TN*	*FP*
Actual Positive	*FN*	*TP*

2) ACCURACY

The *accuracy* is defined as the ratio of correctly predicted observations (both true positives and true negatives) to the total number of observations. It is given by the following formula:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

3) FALSE POSITIVE RATE

The *False Positive Rate (FPR)*, also known as the *Fall-out*, measures the proportion of actual negatives (fake logos) that are incorrectly classified as positives (authentic logos). It reflects how often a model generates false alarms. A high *FPR* indicates that the model is prone to incorrectly identifying fake logos as authentic, which is undesirable in applications where false positives carry a high cost. The *FPR* is defined as:

$$FPR = \frac{FP}{FP + TN} \tag{5}$$

4) FALSE NEGATIVE RATE

The *False Negative Rate (FNR)*, also known as the *Miss Rate*, measures the proportion of actual positives (authentic logos) that are incorrectly classified as negatives (fake logos). A high *FNR* indicates that the model is failing to detect true positives, which could lead to many authentic logos being wrongly classified as fake. The *FNR* is defined as:

$$FNR = \frac{FN}{FN + TP} \tag{6}$$

5) TRUE NEGATIVE RATE

The *True Negative Rate (TNR)*, also known as *Specificity*, measures the proportion of actual negatives (fake logos) that are correctly classified as negatives. A high *TNR* indicates that the model is effective at distinguishing fake logos from authentic ones. The *TNR* is defined as:

$$TNR = \frac{TN}{TN + FP} \tag{7}$$

D. EVALUATION METHODOLOGY

As previously mentioned in this paper, in our system, the first SNN is designed to effectively differentiate between the colors of the logos, while the second SNN focuses on distinguishing the patterns of logos when the logos share similar colors. This second step is crucial for accurate verification when dealing with logos of the same color but printed by different printers. The second model, in this two-step approach, examines the positive sample (both *FP*s and *TP*s) to ensure that they are indeed authentic by verifying the logos' patterns, thereby reducing the error rate. To evaluate the effectiveness of each authentication step, we use the confusion matrices of the first SNN model, the second SNN model, and the final confusion matrix after the overall two-step authentication process. Confusion matrices are detailed after majority voting. The accuracy values indicate the final accuracy after the two-step authentication process with majority voting for $64 \times 64 \times 3$ logo block inputs.

In the closed-set tests, where authentic test logos belong to the same classes (same colors) used for training, we are interested in evaluating the *FPR* and *FNR* of the first model and after the two-step authentication process. This analysis was chosen to reveal how our two-step authentication method reduces the error rates in the closed-set case. For these experiments, we keep the similarity decision threshold equal to 0.5 in both networks, as this value is commonly used in verification with Siamese Networks. Such a testing environment is very simple, does not require setting ad-hoc thresholds, and has already been used before in literature [6].

For the open-set tests, we conduct experiments with various threshold pairs to study their effects in a completely unknown scenario. In such experiments, we use the *TNR* and *FNR*. We report the *TNR* and *FNR* values for all examined threshold pairs. We chose to evaluate only negatives (*fakes*) classification/misclassification in the open-set scenario, as our concerns are related to minimizing cases where an unknown fake logo with the same color as the authentic new logo is misclassified as pristine while minimizing the detection of unknown fakes (*TNR*). We are also interested in reducing the *FNR*, a metric that is maximized when a pristine new logo is misclassified as fake.

In the open-set case, the error rates depend on two thresholds (determining the outcome of the first and second SNNs); for this reason, it is not possible to directly plot the *TNR* as a function of the *FNR*. Rather, by plotting the (*TNR*, *FNR*) pairs for different threshold pairs, we obtain a cloud of points, whose upper envelope indicates the best achievable *TNR* for a given *FNR* (as usually done by ROC curves) and the corresponding threshold pairs. This visualization also helps understand how different threshold settings impact the model's ability to correctly classify authentic and fake logos.

V. EXPERIMENTAL RESULTS

In this section, we describe the results we got with the various architectures we have proposed. We divide this section into three subsections of experiments: closed-set, open-set, and model sizes. We discuss each of them in the following.

A. CLOSED-SET EXPERIMENTS

For the closed-set experiments, we compare our proposed networks in a scenario where the authentic logos have a known color (that is, a color used for training), and fakes are unknown to the classifier. We used a threshold of 0.5 for detecting fake or authentic logos as explained previously in Section IV.

1) MCS−SNN RESULTS

We start analyzing the overall accuracies at the logo level, that is, after majority voting, of the MCS−SNN on closed-set samples. Each row in the Table 8 corresponds to a different reference logo, while each column corresponds to fake logos and authentic logos used for testing. For instance, the (D.O, (D.O, Fakes)) element of the table reports the total accuracy we got when pairing an authentic Dark Orange template with an authentic Dark Orange and fakes. The accuracies reported in the table are the overall accuracies obtained on both authentic and fake pairs (a breakdown of the various error probabilities is discussed later). For this specific network, we got a mean accuracy of 99.27% considering the 12 experiments reported in the Table 8. The network reached accuracies higher than 97% even when an authentic logo is compared against an authentic template with a different color (*e.g.*, when an authentic Dark Orange template is compared against an authentic Light Orange logo).

TABLE 8. Closed-set authentication accuracies using the MCS−SNN model. Each entry shows the accuracy of verifying a reference authentic logo (*e.g.*, Dark Orange) against a set containing the same-color authentic logos and corresponding fakes. Cross-color authentic comparisons are also included. Logo pairs are accepted as authentic only if they match in color and printer origin.

Closed-set - Accuracy (MCS−SNN)				
	Test data			
Template	*(D.O, Fakes)*	*(L.O, Fakes)*	*(L.B, Fakes)*	*(D.B, Fakes)*
D.O	97.37%	98.57%	98.61%	98.03%
L.O	99.74%	98.76%	99.63%	99.83%
L.B	99.94%	99.88%	98.40%	99.67%
D.B	99.99%	99.98%	99.97%	100.00%

TABLE 9. Closed-set confusion matrices using the MCS−SNN architecture. Each entry shows confusion matrices from the first step (top), the second step (middle), and the combined two-step system (bottom). Rows represent the reference logo template; columns represent test logos. Yellow cells indicate improvements from the second step, while bold entries highlight key results discussed in the text. Results reflect logo-level classification after majority voting.

Confusion Matrices - Closed-set (MCS−SNN)				
Template	*(D.O, Fakes)*	*(L.O, Fakes)*	*(L.B, Fakes)*	*(D.B, Fakes)*
D.O	5064 178 59 5183 32 146 70 5113 5096 146 129 5113	15605 293 0 0 67 226 0 0 15672 226 0 0	8308 137 0 0 20 117 0 0 8328 117 0 0	597 13 0 0 1 12 0 0 598 12 0 0
L.O	10390 94 0 0 67 27 0 0 10457 27 0 0	7827 122 71 7878 84 38 87 7791 7911 38 158 7791	8377 68 0 0 37 31 0 0 8414 31 0 0	607 3 0 0 2 1 0 0 609 1 0 0
L.B	10444 40 0 0 34 6 0 0 10478 6 0 0	15823 75 0 0 57 18 0 0 15580 18 0 0	4189 33 3 4220 26 7 125 4095 4215 7 128 4095	302 308 0 0 **306** 2 0 0 608 2 0 0
D.B	10447 37 0 0 36 1 0 0 10483 1 0 0	15825 73 0 0 70 3 0 0 15895 3 0 0	4194 4251 0 0 **4249** 2 0 0 8443 2 0 0	302 3 0 305 3 0 0 305 305 0 0 305

Table 9 shows the confusion matrices of the first, second, and two-step authentication, highlighting misclassification fixes. As a representative example, we see that the accuracy achieved by the first model of MCS−SNN is only 50% when a light blue authentic template is compared with dark blue authentic and fakes templates (top confusion matrix of the (L.B, (D.B, Fakes)) entry in Table 9). When a light blue random template is paired with dark blue authentic test logos, the initial model classifies most of the dark blue logos as authentic, but the second model (whose results are highlighted in bold in Table 9 at (L.B, (D.B, Fakes)) entry) turns most of these errors to true decisions, yielding a final accuracy of 99.67%. A similar effect from the two-step authentication also helps to achieve good accuracy when a Dark Blue template is tested against Light Blue authentic logos and fake logos (entry (D.B, (L.B, Fakes) in Table 9).

TABLE 10. *False Positive Rate* and *False Negative Rate* before and after two-step authentication using the MCS−SNN architecture. The first row shows results from the initial model, and the second row shows results after applying the two-step approach. Yellow highlights indicate significant improvements in error reduction.

False Positive Rate (FPR) and False Negative Rate (FNR) - Closed-set (MCS−SNN)								
Template	*(D.O, Fakes)*		*(L.O, Fakes)*		*(L.B, Fakes)*		*(D.B, Fakes)*	
D.O	FPR(%) FNR(%) 3.36 1.12 2.76 2.47		FPR(%) FNR(%) 1.85 - 1.42 -		FPR(%) FNR(%) 1.62 - 1.37 -		FPR(%) FNR(%) 2.14 - 1.97 -	
L.O	FPR(%) FNR(%) 0.89 - 0.25 -		FPR(%) FNR(%) 1.54 0.89 0.48 1.98		FPR(%) FNR(%) 0.8 - 0.36 -		FPR(%) FNR(%) 0.49 - 0.16 -	
L.B	FPR(%) FNR(%) 0.38 - 0.06 -		FPR(%) FNR(%) 0.47 - 0.11 -		FPR(%) FNR(%) 0.78 0.07 0.16 3.06		FPR(%) FNR(%) 50.49 - 0.33 -	
D.B	FPR(%) FNR(%) 0.35 - 0.01 -		FPR(%) FNR(%) 0.46 - 0.02 -		FPR(%) FNR(%) 50.37 - 0.02 -		FPR(%) FNR(%) 0.98 0 0 0	

In the Table 10, we report the same results by turning them into *FPR* and *FNR* to show the effect of the two-step procedure in reducing them. In particular, we report the rates

at the output of the first authentication step (top of each matrix) and the end of the two steps (bottom of each matrix), revealing the noticeable error reduction brought by the second verification step. By observing the Table 10, we see that in some instances *FPR* is substantially reduced by the fine-tuned model. The most remarkable gains are highlighted in yellow in the table. At the same time, the *FNR* increases only slightly. We observe that the choice to minimize *FNR* or *FPR* depends on the specific problem and the consequences of such errors. In our application, we prioritized reducing one type of error (*FPR*) over the other.

In conclusion, the shallow-and-wide MCS−SNN architectures exhibit very good accuracies and perform well in closed-set conditions.

2) MCC−SNN

In this section, we repeat the analysis described in Section V-A1 for the MCC−SNN network (see Table 11). Even in this case, the network achieves a remarkable result in the closed-set scenario. Considering the 12 experiments, the MCC−SNN network obtains a mean accuracy of 99.56%, with 6 entries showing perfect authentication.

TABLE 11. Closed-set authentication accuracies using the MCC−SNN model. Results in this table were created using the same experimental setup as in Table 8, comparing reference authentic logos with same or different-color samples and fakes to evaluate within-printer consistency.

Closed-set - Accuracy (MCC−SNN)				
	Test data			
Template	*(D.O, Fakes)*	*(L.O, Fakes)*	*(L.B, Fakes)*	*(D.B, Fakes)*
D.O	97.48%	99.29%	99.27%	99.67%
L.O	99.87%	97.74%	99.84%	100.00%
L.B	99.99%	99.99%	99.89%	100.00%
D.B	100.00%	100.00%	100.00%	100.00%

TABLE 12. Closed-set confusion matrices using the MCC−SNN architecture. Format and interpretation are the same as in the Table 9. Improvements from the second step are marked in yellow; key observations are in bold. All results are at the logo level after majority voting.

Confusion Matrix - Closed-set (MCC−SNN)				
Template	*(D.O, Fakes)*	*(L.O, Fakes)*	*(L.B, Fakes)*	*(D.B, Fakes)*
D.O	5013 229 161 5081 143 86 17 5064 5156 86 178 5064	8906 6992 0 0 **6880** 112 0 0 15786 112 0 0	8268 177 0 0 116 61 0 0 8384 61 0 0	600 10 0 0 8 2 0 0 608 2 0 0
L.O	5904 4580 0 0 **4567** 13 0 0 10471 13 0 0	7716 233 323 7626 213 20 15 7611 7929 20 338 7611	8314 131 0 0 118 13 0 0 8432 13 0 0	602 8 0 0 8 0 0 0 610 0 0 0
L.B	10294 190 0 0 189 1 0 0 10483 1 0 0	15617 281 0 0 280 1 0 0 15897 1 0 0	4049 173 4 4219 172 1 4 4215 4221 1 8 4215	293 317 0 0 317 0 0 0 610 0 0 0
D.B	10318 166 0 0 166 0 0 0 10484 0 0 0	15647 251 0 0 251 0 0 0 15898 0 0 0	4071 4374 0 0 **4374** 0 0 0 8445 0 0 0	294 11 0 305 11 0 0 305 305 0 0 305

Based on the analysis of the confusion matrices shown in Table 12, we show one example where the initial MCC−SNN model exhibits an accuracy of 56% when comparing a dark orange authentic template with a light orange and fakes (first confusion matrix at (D.O,(L.O, Fakes)) entry). Notably, when a dark orange random template is compared against a test set with light orange authentics, it is expected that all of them should be classified solely as fakes. However, the initial model erroneously categorizes most of the light orange logos as authentic. In contrast, the subsequent model (highlighted in bold in Table 12 at (D.O, (L.O, Fakes)) entry), having acquired discriminative feature recognition, effectively identifies nearly all FPs as TNs, showing a final accuracy of 99.29% after the two authentication steps for (D.O, (L.O, Fakes)) experiment. The incorporation of a two-step authentication process significantly enhances accuracy when evaluating other pairs in that table, for example, the (L.O, (D.O, Fakes)), (L.B, (DB, Fakes)), and (D.B, (L.B, Fakes)), as we can see highlighted in bold in Table 12.

TABLE 13. *False Positive Rate* and *False Negative Rate* before and after two-step authentication using the MCC−SNN architecture. The first row corresponds to the initial model, and the second to the two-step system. Yellow cells mark notable reductions in error rates due to the second step.

False Positive Rate (FPR) and False Negative Rate (FNR) - Closed-set (MCC−SNN)								
Template	*(D.O, Fakes)*		*(L.O, Fakes)*		*(L.B, Fakes)*		*(D.B, Fakes)*	
D.O	FPR(%)	FNR(%)	FPR(%)	FNR(%)	FPR(%)	FNR(%)	FPR(%)	FNR(%)
	4.32	3.09	43.93	-	2.11	-	1.63	-
	1.64	3.32	0.70	-	0.72	-	0.33	-
L.O	FPR(%)	FNR(%)	FPR(%)	FNR(%)	FPR(%)	FNR(%)	FPR(%)	FNR(%)
	43.63	-	2.97	4.09	1.56	-	1.82	-
	0.12	-	0.25	4.23	0.15	-	0	-
L.B	FPR(%)	FNR(%)	FPR(%)	FNR(%)	FPR(%)	FNR(%)	FPR(%)	FNR(%)
	1.82	-	1.78	-	4.09	0.09	51.93	-
	0.01	-	0.01	-	0.02	0.19	0	-
D.B	FPR(%)	FNR(%)	FPR(%)	FNR(%)	FPR(%)	FNR(%)	FPR(%)	FNR(%)
	1.59	-	1.58	-	51.73	-	3.67	0
	0	-	0	-	0	-	0	0

The analysis of the results in Table 13 highlights the effectiveness of our two-step authentication strategy. By looking at the FPR and FNR reported in the table, we see a substantial decrease in FPRs passing from 43.93% to 0.79%, 43.63% to 0.12%, 51.93% to 0%, and 51.73% to 0% (highlighted in yellow in Table 13). Notably, the initial model's errors were significantly curbed by the two-step authentication.

3) MINI-XCEPTION

Finally, we analyse the results obtained by Mini−Xception. With its shallow but efficient design, Mini−Xception exhibits superior feature extraction capabilities, enabling it to distinguish subtle differences in printed logos. As can be seen in Table 14, it outperforms the shallow-and-wide architectures in the closed-set scenario, having 100% accuracy in 7 entries of the table, and has the highest mean accuracy in the closed-set environment (99.82%).

Some additional insights into the performance of the Mini−Xception network can be obtained by looking at Table 15. In particular, we observe how the two-step authentication does not improve much the accuracy of the first step, being the complexity of the network is enough to detect even hard fakes in the first step. The depthwise separable convolutions and increased model capacity of the

TABLE 14. Closed-set authentication accuracies using the `Mini-Xception` model. Accuracy is computed by comparing each reference logo with matching and non-matching authentic samples and fakes. As with previous models, only logos printed in the same color by the same printer should be accepted.

Closed-set - Accuracy (`Mini-Xception`)				
	Test data			
Template	*(D.O, Fakes)*	*(L.O, Fakes)*	*(L.B, Fakes*	*(D.B, Fakes)*
D.O	99.42%	99.73%	99.73%	99.67%
L.O	99.96%	98.91%	99.96%	100.00%
L.B	100.00%	99.99%	99.83%	100.00%
D.B	100.00%	100.00%	100.00%	100.00%

TABLE 15. Closed-set confusion matrices using the `Mini-Xception` architecture. Same structure and analysis criteria as previous confusion matrix tables. Each entry includes first-step, second-step, and combined results. Highlighting indicates performance gains and relevant comparisons.

Confusion Matrix - Closed-set (`Mini-Xception`)				
Template	*(D.O, Fakes)*	*(L.O, Fakes)*	*(L.B, Fakes)*	*(D.B, Fakes)*
D.O	5180 62 2 5240	15816 82 0 0	8406 39 0 0	603 7 0 0
	27 35 23 5217	40 42 0 0	17 22 0 0	5 2 0 0
	5207 35 25 5217	15856 42 0 0	8426 2 0 0	608 2 0 0
L.O	10463 21 0 0	7914 35 4 7945	8429 16 0 0	609 1 0 0
	17 4 0 0	30 5 164 7781	13 3 0 0	1 0 0 0
	10480 4 0 0	7944 5 168 7781	8442 3 0 0	610 0 0 0
L.B	10484 0 0 0	15894 4 0 0	4222 0 0 4223	305 305 0 0
	0 0 0 0	3 1 0 0	0 0 14 4209	**305** 0 0 0
	10484 0 0 0	15897 1 0 0	4222 0 14 4209	610 0 0 0
D.B	10484 0 0 0	15894 4 0 0	4222 4223 0 0	305 0 0 305
	0 0 0 0	4 0 0 0	**4223** 0 0 0	0 0 0 305
	10484 0 0 0	15898 0 0 0	8445 0 0 0	305 0 0 305

`Mini-Xception` architecture contribute significantly to its excellent performance in the authentication of printed logos, especially in the first step. To reinforce such arguments, it could be seen from Table 15 that, from the 10 highlighted confusion matrices in the second step, only in two of them (entries (D.B, (L.B, Fakes)) and (L.B, (D.B, Fakes)) in Table 15), the second step improves significantly the accuracy.

TABLE 16. The *False Positive Rate* and *False Negative Rate* before and after two-step authentication using the `Mini-Xception` architecture. The top row shows initial model performance, and the bottom row reflects the effect of the two-step strategy. Yellow highlights indicate where the second step reduces error rates most effectively.

False Positive Rate (FPR) and False Negative Rate (FNR))								
Template	*(D.O, Fakes)*		*(L.O, Fakes)*		*(L.B, Fakes)*		*(D.B, Fakes)*	
D.O	FPR(%)	FNR(%)	FPR(%)	FNR(%)	FPR(%)	FNR(%)	FPR(%)	FNR(%)
	1.18	0.03	0.51	-	0.46	-	1.15	-
	0.66	0.48	0.26	-	0.26	-	0.33	-
L.O	FPR(%)	FNR(%)	FPR(%)	FNR(%)	FPR(%)	FNR(%)	FPR(%)	FNR(%)
	0.20	-	0.43	0.05	0.18	-	0.163	-
	0.03	-	0.063	2.12	0.03	-	0	-
L.B	FPR(%)	FNR(%)	FPR(%)	FNR(%)	FPR(%)	FNR(%)	FPR(%)	FNR(%)
	0	-	0.02	-	0	0	50	-
	0	-	0.006	-	0	0.33	0	-
D.B	FPR(%)	FNR(%)	FPR(%)	FNR(%)	FPR(%)	FNR(%)	FPR(%)	FNR(%)
	0	-	0.02	-	50	-	0	0
	0	-	0	-	0	-	0	0

The same results expressed in terms of *FPR* and *FNR* can be found in Table 16. In line with our previous arguments, there is some decrease in *FPR*s and *FNR*s, but the two-step authentication only significantly lowers the number of false positive errors from 50% to 0% in the cases highlighted in yellow in the table.

We can explain the outstanding performance of the first step of the `Mini-Xception` system by analyzing its loss curves we presented previously in this paper. Figure 11 (a) shows that the first step SNN `Mini-Xception` stabilizes both the accuracy and loss faster and better than the other networks (Figures 10 (a) and 9 (a)). However, an opposite effect is observed in the second step (11 (b)). The reason for this behaviour is that when a limited amount of data is available (as it is in the hard fakes dataset used to train the second step model), `Mini-Xception`'s complex architecture could not learn the particularities as well as the other networks (Figures 10 (b) and 9 (b))). Such a limitation will make the `Mini-Xception` model pay a high price in the open-set scenario, as we will see in the next subsection.

B. OPEN-SET EXPERIMENTS

In this section, we evaluate the system's robustness and adaptability to new, unforeseen situations through open-set tests. Such tests involve training the model on a subset of known classes and then introducing unseen classes during the evaluation phase. The goal is to determine how well the system performs when an input belongs to a class not used during training.

1) DATASET AND METHODOLOGY

We conducted the open-set experiments by organizing an unknown evaluation dataset into two groups, containing unknown samples for authentic (green logos) and fake logos. This evaluation setup allows for testing the model's ability to recognize authentic unknown classes while rejecting instances from fake unseen classes, simulating situations where systems encounter data outside the training distribution. We used 7,536 authentic green logos and an equal number of fake logos for our experiments to ensure a balanced evaluation. These logos were not included in the training phase of any of the networks discussed earlier. To authenticate the logos in the open-set scenario, we select one random authentic unknown logo (green) as the template for each test sample. The test sample and the template are then compared by the SNNs in the two-step authentication procedure.

In open-set scenarios, the overall optimization of the system may require that the decision thresholds of the first and second authentication steps be jointly set in an optimum way. To understand how to do so, we plotted the *FNR* and *TNR* for various values of the two thresholds, obtaining a cloud of points in the (*FNR*; *TNR*) plane. The upper envelope of the cloud gives the best achievable *TNR* when the *FNR* is fixed. The envelope provides a visual representation of the overall performance boundary across the decision thresholds of the two networks that the authentication system consists of.

2) RESULTS

a: MCS−SNN RESULTS

We evaluated the performance of the MCS−SNN verification system under open-set scenarios across various criteria. In Figure 12 we report the envelope of the achievable (*TNR*, *FNR*) pairs. Then, in Table 17 we report the results for some pairs of thresholds of special interest, namely the highest accuracy achieved across all threshold pairs we tested, followed by the accuracy at the standard threshold pair of (0.5, 0.5). Next, we report the model's accuracy under application-specific constraints where the *FNR* is restricted to be less than or equal to 5%. As shown in the table, the MCS−SNN architecture achieves its highest accuracy of 83.41% using the threshold pair (0.2, 0.05), showcasing its ability to perform well under ideal conditions. However, when the standard threshold pair (0.5, 0.5) is applied, the accuracy slightly decreases to 81.17%, indicating a marginal trade-off for the standard (0.5, 0.5) pair of thresholds. Under the application-specific constraint where the *False Negative Rate (FNR)* must be less than 5%, the accuracy drops further to 72.53%, achieved with the threshold pair (0.1, 0.85).

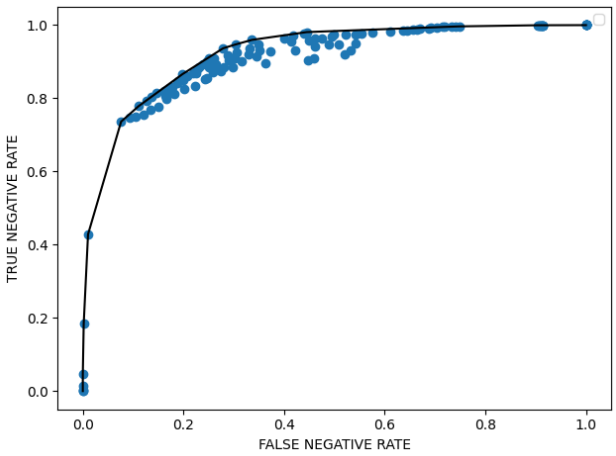


FIGURE 12. ***TNR*** **and** ***FNR*** **values achieved by the** MCS−SNN **architecture across various threshold pairs. The envelope of the plot represents the best achievable performance trade-off between false negatives and true negatives.**

TABLE 17. Open-set performance of the MCS−SNN **model across selected threshold pairs. The table reports accuracy, *TNR*, and *FNR* for three threshold combinations: (i) the pair yielding the highest overall accuracy, (ii) the standard baseline (0.5, 0.5), and (iii) a configuration where *FNR* remains below 5%. These points illustrate the trade-offs between rejecting fake logos and retaining authentic ones in open-set conditions.**

MCS-SNN Model					
Criteria	*T1*	*T2*	*Accuracy*	*TNR*	*FNR*
Best Accuracy	0.2	0.05	83.41%	77.87%	11.09%
Standard Threshold Pair	0.5	0.5	81.17%	95.97%	33.62%
FNR $< 5\%$	0.1	0.85	72.53%	90.92%	4.58%

b: MCC−SNN RESULTS

Concerning the MCC−SNN architecture, from Figure 13 and the thresholds of interest reported in Table 18, we can state that the MCC−SNN architecture outperforms MCS−SNN, with a peak accuracy of 89.66% at the threshold pair (0.2, 0.05), reflecting its strong capabilities under optimal conditions. When evaluated with the standard threshold pair (0.5, 0.5), the accuracy slightly decreases to 87.7%, but it remains higher than that of MCS−SNN. Under the constraint of $FNR < 5\%$, the MCC−SNN demonstrates a remarkable stability with an accuracy of 87.86% at the threshold pair of (0.1, 0.15), indicating a minimal loss in performance.

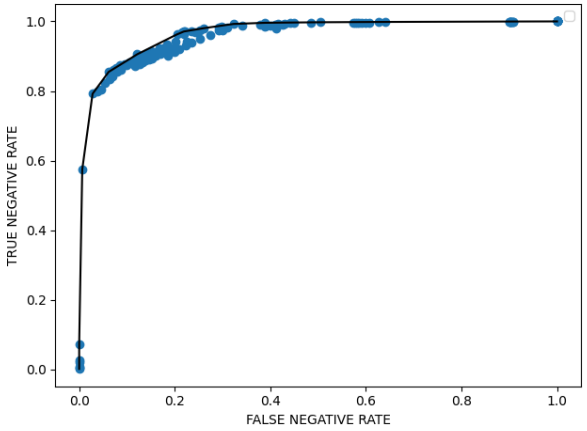


FIGURE 13. ***TNR*** **and** ***FNR*** **values achieved by the** MCC−SNN **architecture across various threshold pairs. The performance envelope outlines the optimal balance between true negative rate and false negative rate.**

TABLE 18. Open-set performance of the MCC−SNN **model across selected threshold pairs. Evaluation metrics include accuracy, *TNR*, and *FNR* for three representative threshold settings: best overall accuracy, standard baseline (0.5, 0.5), and *FNR* constrained below 5%. These results highlight the balance between minimizing false positives and maintaining low false negatives in open-set verification.**

MCC-SNN Model					
Criteria	*T1*	*T2*	*Accuracy*	*TNR*	*FNR*
Best Accuracy	0.2	0.05	89.66%	85.5%	6.17%
Standard	0.5	0.5	87.7%	97.72%	22.28%
FNR $< 5\%$	0.1	0.15	87.86%	80.4%	4.67%

c: MINI−XCEPTION RESULTS

As can be seen in Figure 14 and Table 19, the Mini−Xception architecture exhibits the worst performance in open-set conditions among the three models. Its highest accuracy, 68.77%, is achieved using the threshold pair (0.9, 0.1), which highlights its limitations in the two-step authentication under open-set conditions. At the standard threshold pair (0.5, 0.5), the accuracy further declines to 67.1%, emphasizing its reduced effectiveness compared to the other architectures. Notably, across all tested threshold pairs, Mini−Xception fails to satisfy the *FNR* less than 5% constraint, reflecting its limitations in handling stricter application-specific requirements.

d: DISCUSSION

To conclude our analysis, we notice that in the closed-set scenario, Mini−Xception achieved slightly better results compared to the other models. However, in the open-set

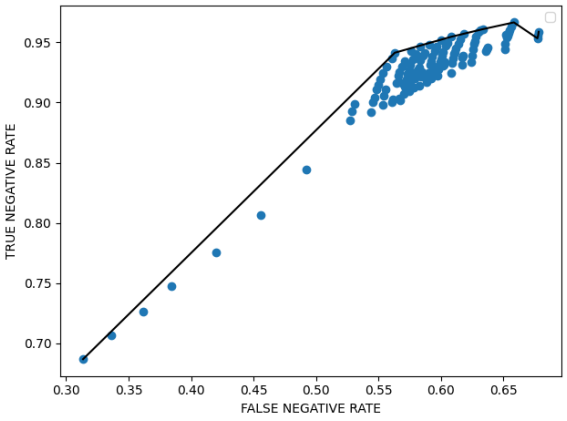


FIGURE 14. ***TNR*** **and** ***FNR*** **values achieved by the** `Mini-Xception` **architecture across various threshold pairs. The envelope reflects the upper bound of achievable performance in minimizing false negatives while maintaining high true negative rates.**

TABLE 19. Open-set performance of the `Mini-Xception` **model across selected threshold pairs. The table shows accuracy, *TNR*, and *FNR* for the best-performing threshold and the standard baseline (0.5, 0.5). No configuration met the *FNR* < 5% criterion, reflecting the limitations of lightweight models in high-security open-set scenarios.**

Mini-Xception Model					
Criteria	*T1*	*T2*	*Accuracy*	*TNR*	*FNR*
Best Accuracy	0.9	0.1	68.77%	93.65%	56.11%
Standard	0.5	0.5	67.1%	93.43%	59.22%
$FNR < 5\%$	-	-	-	-	-

scenario, the first step of `Mini-Xception` overfits to unknown pristine (green logos) contained in the open-set, and the second step also fails to correct errors from the first step (that means, a pristine green authentic unknown logo to the classifier is classified as fake by both steps of `Mini-Xception`). This overfitting highlights that `Mini-Xception` is overly complex and suffers from high variance, making it less effective for open-set tasks when an authentic new logo is considered (as it can be seen from their high False Negative Rates in Table 19). In contrast, a simpler model like `MCC-SNN`, with its well-trained two-step process, performs significantly better in open-set scenarios. Overall, `MCC-SNN` demonstrates superior performance compared to both `MCS-SNN` and `Mini-Xception` across all evaluated criteria. It achieves the highest top accuracy, maintains robust performance under the standard threshold, and meets the $FNR < 5\%$ constraint with minimal loss in accuracy. While `MCS-SNN` performs reasonably well in the open and closed sets, it falls short of `MCC-SNN` in overall metrics. On the other hand, `Mini-Xception` exhibits in the open-set the lowest accuracy, fails to meet the *FNR* constraint in any threshold pair, and struggles with open-set classification due to its high complexity and overfitting tendencies.

C. BASELINE COMPARISON

1) ARCHITECTURES USED

To rigorously assess the performance and robustness of our proposed authentication framework, we conducted an extensive comparative analysis using four prominent state-of-the-art DL architectures as baselines: ResNet-50 [64], EfficientNet-B0 [65], DenseNet-121 [66], and MobileNet-V2 [67]. These architectures were deliberately chosen based on their widespread adoption, proven performance across diverse classification tasks, and varied structural complexities, thereby providing a comprehensive evaluation landscape.

ResNet-50 employs residual learning with skip connections, enabling the effective training of deeper networks. EfficientNet-B0 is notable for its balance between computational efficiency and accuracy, leveraging compound scaling techniques to optimize model performance. DenseNet-121 is characterized by dense connectivity patterns that facilitate feature reuse and reduce gradient vanishing problems. MobileNet-V2 is designed specifically for efficiency and speed, incorporating depthwise separable convolutions that significantly reduce computational overhead.

2) DATASET AND EXPERIMENTAL SETUP

For a fair and controlled comparison, all baseline models were trained and evaluated using precisely the same dataset and experimental conditions as our proposed Siamese-based approach. The dataset consisted of printed logo images captured from various printers, ensuring sufficient variability to simulate realistic conditions. Moreover, we maintained an identical number of images in the training, validation, and test splits across all experiments, thereby guaranteeing consistency in dataset partitioning.

To comply with the standardized input dimensions expected by the baseline architectures, all images were resized to $(224 \times 224 \times 3)$ pixels. In contrast, our proposed patch-based Siamese verification system processes smaller patches of size $(64 \times 64 \times 3)$ and aggregates predictions through majority voting. This architectural choice highlights a critical divergence from traditional image classification pipelines, explicitly underscoring the strengths of our patch-based similarity verification strategy.

3) EVALUATION IN OPEN-SET CONDITIONS

The baseline evaluation was intentionally conducted under open-set conditions to assess the generalization capabilities of the models in scenarios that closely resemble real-world deployments. Given that closed-set evaluations, where the test data includes only logos and printers encountered during training, typically yield high accuracy, we deliberately adopted a more challenging open-set setting. This approach ensures that the evaluation rigorously reflects the models' performance in realistic environments characterized by the frequent emergence of novel logo-printer combinations. Our models and baseline model performances were evaluated using a default decision threshold of 0.5, a standard choice for binary verification problems.

4) COMPARATIVE RESULTS

Our comparative analysis revealed that all baseline models substantially underperformed relative to our proposed

Siamese-based approach. Specifically, the baseline models struggled to maintain robust performance in open-set scenarios, often misclassifying unseen logo-printer combinations. In contrast, our two-step SNN framework consistently demonstrated superior discriminative capabilities and robustness, effectively distinguishing authentic from fraudulent logos even when encountering previously unseen combinations.

These findings are succinctly summarized in Table 20, where baseline Siamese models using full-logo inputs demonstrated limited generalization under open-set conditions. For instance, `ResNet-50` achieved only 47.15% accuracy with a *TNR* of 64.42% and a high *FNR* of 70.11%. Similarly, `EfficientNet-B0` showed particularly poor performance with just 46.05% accuracy, an extremely low *TNR* of 8.38%, and an *FNR* of 41.49%. While `DenseNet-121` slightly outperformed others with 57.90% accuracy and a *TNR* of 62.10%, its *FNR* remained relatively high at 46.29%. `MobileNet-V2`, despite a high *TNR* of 97%, failed to generalize effectively, as evidenced by its modest 49.85% accuracy and an alarmingly high *FNR* of 97.30%.These results clearly illustrate the superior generalization capability of our proposed Siamese-based approach, emphasizing the effectiveness of similarity-based learning in handling fine-grained verification tasks under open-set conditions mirroring the real-world scenarios.

TABLE 20. Open-set performance summary of baseline SNN models against our proposed models. Accuracy, *TNR*, and *FNR* are reported for all CNN-based Siamese networks. Results are computed at the standard similarity threshold of 0.5.

SNN Model	*Accuracy*	*TNR*	*FNR*
ResNet-50[64]	47.15%	64.42%	70.11%
EfficientNet B0[65]	46.05%	8.38%	41.49%
DenseNet 121[66]	57.90%	62.10%	46.29%
MobileNet V2[67]	49.85%	97%	97.30%
MCS-SNN	81.17%	95.97%	33.62%
MCC-SNN	87.7%	97.72%	22.28%
Mini-Xception	67.1%	93.43%	59.22%

The observed superior open-set performance of our Siamese-based framework underscores its suitability for practical deployment in real-world scenarios, where novel logos and printing devices continuously emerge. Unlike traditional classification models, which may require extensive retraining to adapt to new logo-printer combinations, our proposed method inherently supports generalization to novel data, significantly reducing the need for frequent model updates.

In conclusion, this qualitative state-of-the-art comparison decisively demonstrates the robustness and efficacy of our proposed patch-based Siamese verification approach, establishing it as a highly advantageous solution for printed logo authentication tasks.

D. MODELS SIZES

An important advantage of the proposed approach is its computational efficiency and compact model size. Table 21 summarizes the size (in Megabytes) of all SNN models evaluated in this work. For the proposed architectures, namely `MCS-SNN`, `MCC-SNN`, and `Mini-Xception`, we report the model sizes for both the initial training stage (from scratch) and the fine-tuned second stage, which together constitute the two-step authentication pipeline. In contrast, the baseline models based on popular deep CNN architectures, including `ResNet-50`, `EfficientNet-B0`, `DenseNet-121`, and `MobileNet-V2`, are evaluated in a single-step setting using full-logo images, as they are not integrated into the two-step authentication framework. Their model sizes are also reported in Table 21 for comparative purposes.

TABLE 21. Model size comparison for proposed and baseline SNN architectures. Sizes (in MB) are shown for proposed models in both steps of the two-step pipeline, and baselines used a single-step full-logo setting.

Models' size (in MBytes)		
SNN model	*SNN training method*	
	First step	*Second Step*
MCS-SNN	0.899	0.898
MCC-SNN	1.1	1.1
Mini-Xception	25.5	25.5
ResNet-50[64]	99	-
EfficientNet-B0[65]	19.4	-
DenseNet-121[66]	31	-
MobileNet-V2[67]	12.1	-

The results clearly highlight the lightweight nature of most of our proposed architectures (in particular `MCS-SNN` and `MCC-SNN`). The combined size of all three proposed models remains under 56MB, making them not only suitable for efficient training and evaluation in cloud environments but also viable for deployment on memory-constrained devices such as smartphones. Furthermore, the modularity of the two-step design allows flexible system customization: for instance, using `Mini-Xception` for robust detection of known authentic logos in the first step, followed by `MCC-SNN` for improved discrimination against high-quality fakes in the second step. This architecture, along with threshold tuning, opens further avenues for fine-grained anti-counterfeiting solutions deployable in real-world conditions.

VI. CONCLUSION AND FUTURE WORK

In this paper, we introduced a practical and effective solution for printed logo authentication as a means to combat the growing issue of counterfeit goods. We validated our methods by using a dataset of logo patches of five different colors and a huge set of fake logos, both in closed-set and open-set conditions. By formulating the authentication process as a printed logo verification task, we proposed a DL-based system that leverages the subtle chromatic and geometric artifacts introduced by different printers. Our solution consists of a two-step authentication pipeline based on cascaded SNNs. The first SNN detects easily distinguishable counterfeits by exploiting chromatic differences, while the second is fine-tuned to capture subtle geometric patterns, enabling the identification of high-quality fakes.

The proposed system is designed to work under realistic conditions using low-cost acquisition devices and without requiring modifications to the printing process. Furthermore, by employing shallow-and-wide CNNs, our system can be effectively trained on small datasets and has shown strong generalization capabilities under open-set conditions. Experimental results on a challenging dataset demonstrate the system's performance and potential for real-world applications outperforming several baseline models.

Several avenues for future research can be considered to further enhance the system's effectiveness and generalizability. Firstly, a more extensive evaluation on larger and more diverse datasets, covering a broader spectrum of authentic logos and counterfeiting techniques, would provide deeper insights into the system's scalability. Secondly, exploring the combination of different SNN architectures within the two-step verification pipeline could leverage their individual strengths and improve accuracy and robustness. Lastly, the integration of advanced techniques such as adversarial training [68], [69] may further strengthen the system's resilience against informed adversaries attempting to bypass the authentication mechanism. These directions point toward the development of a more comprehensive and secure logo-based authentication framework, reinforcing its applicability in real-world scenarios to safeguard brands and consumers alike.

ACKNOWLEDGMENT

In this article, only parts of the introduction contain excerpts generated with ChatGPT. All other sections were written organically and are the sole responsibility of the authors.

REFERENCES

[1] *A Study on Public Health and Socioeconomic Impact of Substandard and Falsified Medical Products*, World Health Organization, Geneva, Switzerland, 2017.

[2] G. N. Miceli and R. Pieters, "Looking more or less alike: Determinants of perceived visual similarity between copycat and leading brands," *J. Bus. Res.*, vol. 63, no. 11, pp. 1121–1128, Nov. 2010.

[3] M. Gao, J. Li, D. Xia, L. Jiang, N. Peng, S. Zhao, and G. Li, "Lanthanides-based security inks with reversible luminescent switching and self-healing properties for advanced anti-counterfeiting," *J. Mol. Liquids*, vol. 350, Mar. 2022, Art. no. 118559.

[4] H. Zheng, C. Zhou, X. Li, Z. Guo, and T. Wang, "A novel steganography-based pattern for print matter anti-counterfeiting by smartphone cameras," *Sensors*, vol. 22, no. 9, p. 3394, Apr. 2022.

[5] I.-H. Lee, G. Li, B.-Y. Lee, S.-U. Kim, B. Lee, S.-H. Oh, and S.-D. Lee, "Selective photonic printing based on anisotropic Fabry–Perot resonators for dual-image holography and anti-counterfeiting," *Opt. Exp.*, vol. 27, no. 17, pp. 24512–24523, 2019.

[6] A. Ferreira, N. Purnekar, and M. Barni, "Ensembling shallow Siamese neural network architectures for printed documents verification in data-scarcity scenarios," *IEEE Access*, vol. 9, pp. 133924–133939, 2021.

[7] R. A. Merrill, E. G. Bartick, and J. H. Taylor, "Forensic discrimination of photocopy and printer toners I. The development of an infrared spectral library," *Anal. Bioanal. Chem.*, vol. 376, no. 8, pp. 1272–1278, 2003.

[8] D. K. Shaffer, "Forensic document analysis using scanning microscopy," *Proc. SPIE*, vol. 7378, pp. 398–408, May 2009.

[9] G. N. Ali, A. K. Mikkilineni, J. P. Allebach, E. J. Delp, P.-J. Chiang, and G. T. Chiu, "Intrinsic and extrinsic signatures for information hiding and secure printing with electrophotographic devices," in *Proc. Int. Conf. Digit. Printing Technol.*, vol. 19, 2003, pp. 511–515.

[10] I. Tkachenko, W. Puech, C. Destruel, O. Strauss, J.-M. Gaudin, and C. Guichard, "Two-level QR code for private message sharing and document authentication," *IEEE Trans. Inf. Forensics Security*, vol. 11, pp. 571–583, 2016.

[11] Iu. Tkachenko, W. Puech, O. Strauss, C. Destruel, and J.-M. Gaudin, "Printed document authentication using two level or code," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2016, pp. 2149–2153.

[12] I. Tkachenko, A. Trémeau, and T. Fournel, "Authentication of rotogravure print-outs using a regular test pattern," *J. Inf. Secur. Appl.*, vol. 66, May 2022, Art. no. 103133.

[13] Y. Yan, Z. Zou, H. Xie, Y. Gao, and L. Zheng, "An IoT-based anti-counterfeiting system using visual features on QR code," *IEEE Internet Things J.*, vol. 8, no. 8, pp. 6789–6799, Apr. 2021.

[14] C. Chen, M. Li, A. Ferreira, J. Huang, and R. Cai, "A copy-proof scheme based on the spectral and spatial barcoding channel models," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 1056–1071, 2020.

[15] A. Ferreira, C. Chen, and M. Barni, "Fusing multiscale texture and residual descriptors for multilevel 2D barcode rebroadcasting detection," in *Proc. IEEE Int. Workshop Inf. Forensics Security (WIFS)*, Dec. 2021, pp. 1–6.

[16] N. Xie, Q. Zhang, Y. Chen, J. Hu, G. Luo, and C. Chen, "Low-cost anti-copying 2D barcode by exploiting channel noise characteristics," *IEEE Trans. Multimedia*, vol. 23, pp. 3752–3767, 2021.

[17] H. Zheng, C. Zhou, X. Li, T. Wang, and C. You, "Forgery detection for anti-counterfeiting patterns using deep single classifier," *Appl. Sci.*, vol. 13, no. 14, p. 8101, Jul. 2023.

[18] M.-J. Tsai, Y.-C. Lee, and T.-M. Chen, "Implementing deep convolutional neural networks for QR code-based printed source identification," *Algorithms*, vol. 16, no. 3, p. 160, Mar. 2023.

[19] Z. Zheng, H. Zheng, J. Ju, D. Chen, X. Li, Z. Guo, C. You, and M. Lin, "A system for identifying an anti-counterfeiting pattern based on the statistical difference in key image regions," *Expert Syst. Appl.*, vol. 183, Nov. 2021, Art. no. 115410.

[20] S. Ge, Z. Xia, J. Fei, Y. Tong, J. Weng, and M. Li, "A robust document image watermarking scheme using deep neural network," *Multimedia Tools Appl.*, vol. 82, no. 25, pp. 1–24, Oct. 2023.

[21] T. Richter, S. Escher, D. Schönfeld, and T. Strufe, "Forensic analysis and anonymisation of printed documents," in *Proc. 6th ACM Workshop Inf. Hiding Multimedia Secur.*, Jun. 2018, pp. 127–138.

[22] J. Hao, X. Kong, and S. Shang, "Printer identification using page geometric distortion on text lines," in *Proc. IEEE China Summit Int. Conf. Signal Inf. Process. (ChinaSIP)*, Jul. 2015, pp. 856–860.

[23] A. H. Eid, M. N. Ahmed, and E. E. Rippetoe, "EP printer jitter characterization using 2D Gabor filter and spectral analysis," in *Proc. 15th IEEE Int. Conf. Image Process.*, Oct. 2008, pp. 1860–1863.

[24] G. N. Ali, A. K. Mikkilineni, E. J. Delp, J. P. Allebach, P.-J. Chiang, and G. T. Chiu, "Application of principal components analysis and Gaussian mixture models to printer identification," in *Proc. Int. Conf. Digit. Printing Technol.*, vol. 20, 2004, pp. 301–305.

[25] A. K. Mikkilineni, P.-J. Chiang, G. N. Ali, G. T. Chiu, J. P. Allebach, and E. J. Delp, "Printer identification based on texture features," in *Proc. Int. Conf. Digit. Printing Technol.*, vol. 20, 2004, pp. 306–311.

[26] A. K. Mikkilineni, P.-J. Chiang, G. N. Ali, G. T. Chiu, J. P. Allebach, and E. J. Delp, "Printer identification based on graylevel co-occurrence features for security and forensic applications," *Proc. SPIE*, vol. 5681, pp. 430–440, Mar. 2005.

[27] A. K. Mikkilineni, O. Ş. Arslan, P.-J. Chiang, R. Kumontoy, J. P. Allebach, G. T. Chiu, and E. J. Delp, "Printer forensics using SVM techniques," in *Proc. Int. Conf. Digit. Printing Technol.*, vol. 21, 2005, pp. 223–226.

[28] A. K. Mikkilineni, N. Khanna, and E. J. Delp, "Forensic printer detection using intrinsic signatures," *Proc. SPIE*, vol. 7880, pp. 278–288, Feb. 2011.

[29] A. Ferreira, L. C. Navarro, G. Pinheiro, J. A. D. Santos, and A. Rocha, "Laser printer attribution: Exploring new features and beyond," *Forensic Sci. Int.*, vol. 247, pp. 105–125, Feb. 2015.

[30] S. Joshi and N. Khanna, "Single classifier-based passive system for source printer classification using local texture features," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 7, pp. 1603–1614, Jul. 2018.

[31] S. Joshi and N. Khanna, "Source printer classification using printer specific local texture descriptor," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 160–171, 2020.

[32] M. Kumar, S. Gupta, and N. Mohan, "A computational approach for printed document forensics using SURF and ORB features," *Soft Comput.*, vol. 24, no. 17, pp. 13197–13208, Sep. 2020.

[33] H. Jain, S. Joshi, G. Gupta, and N. Khanna, "Passive classification of source printer using text-line-level geometric distortion signatures from scanned images of printed documents," *Multimedia Tools Appl.*, vol. 79, nos. 11–12, pp. 7377–7400, Mar. 2020.

[34] A. Ferreira, E. Nowroozi, and M. Barni, "VIPPrint: Validating synthetic image detection and source linking methods on a large scale dataset of printed documents," *J. Imag.*, vol. 7, no. 3, p. 50, Mar. 2021.

[35] A.-T. Phan-Ho, Q.-T. Nguyen, J. Patrix, and J. Verny, "Source printer identification with microscopic printing using deep learning," *IFAC-PapersOnLine*, vol. 55, no. 10, pp. 1177–1182, 2022.

[36] T. Wang, H. Zheng, C. You, and J. Ju, "A texture-hidden anti-counterfeiting QR code and authentication method," *Sensors*, vol. 23, no. 2, p. 795, Jan. 2023.

[37] O. Bulan, J. Mao, and G. Sharma, "Geometric distortion signatures for printer identification," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Apr. 2009, pp. 1401–1404.

[38] H. Wu, X. Kong, and S. Shang, "A printer forensics method using halftone dot arrangement model," in *Proc. IEEE China Summit Int. Conf. Signal Inf. Process. (ChinaSIP)*, Jul. 2015, pp. 861–865.

[39] S.-J. Ryu, H.-Y. Lee, D.-H. Im, J.-H. Choi, and H.-K. Lee, "Electrophotographic printer identification by halftone texture analysis," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Mar. 2010, pp. 1846–1849.

[40] D.-G. Kim and H.-K. Lee, "Color laser printer identification using photographed halftone images," in *Proc. 22nd Eur. Signal Process. Conf. (EUSIPCO)*, Sep. 2014, pp. 795–799.

[41] D.-G. Kim and H.-K. Lee, "Colour laser printer identification using halftone texture fingerprint," *Electron. Lett.*, vol. 51, no. 13, pp. 981–983, Jun. 2015.

[42] Z. Guo, H. Zheng, C. You, X. Xu, X. Wu, Z. Zheng, and J. Ju, "Digital forensics of scanned QR code images for printer source identification using bottleneck residual block," *Sensors*, vol. 20, no. 21, p. 6305, Nov. 2020.

[43] Z. Guo, S. Wang, Z. Zheng, and K. Sun, "Printer source identification of quick response codes using residual attention network and smartphones," *Eng. Appl. Artif. Intell.*, vol. 131, May 2024, Art. no. 107822.

[44] M.-J. Tsai and T.-M. Chen, "A deep learning approach for QR code based printed source identification," in *Proc. 15th Int. Conf. Signal Process. Commun. Syst. (ICSPCS)*, Dec. 2021, pp. 1–6.

[45] M.-J. Tsai and T.-M. Chen, "A non-bottleneck residual approach for QR code printed source identification," in *Proc. IEEE 8th Int. Conf. Big Data Comput. Service Appl. (BigDataService)*, Aug. 2022, pp. 132–136.

[46] Z.-F. Jiang, Q.-H. Zhang, Y.-C. Wang, Y.-L. Liu, Y.-W. Zhao, Y.-Y. Hao, J.-Y. Xu, X. Yang, and X.-H. Chen, "Prediction of laser printers and cartridges based on three-dimensional profiles via discrimination analysis," *Forensic Sci. Int.*, vol. 363, Oct. 2024, Art. no. 112186.

[47] J. Lee, H. Kim, and T.-Y. Kang, "Classification algorithm using halftone features of counterfeit bills and CNN," *J. Forensic Sci.*, vol. 67, no. 1, pp. 345–352, Jan. 2022.

[48] I. Yuadi, U. Nihaya, and F. D. Pratiwi, "Stardist segmentation to determine the printout characteristics of different printers," in *Proc. Int. Conf. Electr. Eng. Informat. (ICEEI)*, Oct. 2023, pp. 1–5.

[49] I. Yuadi, U. Nihaya, and F. D. Pratiwi, "Watershed segmentation for printed source classification," in *Proc. Int. Conf. Electr. Inf. Technol. (IEIT)*, Sep. 2023, pp. 275–280.

[50] Q.-T. Nguyen, A. Mai, L. Chagas, and N. Reverdy-Bruas, "Microscopic printing analysis and application for classification of source printer," *Comput. Secur.*, vol. 108, Sep. 2021, Art. no. 102320.

[51] A. Ferreira, L. Bondi, L. Baroffio, P. Bestagini, J. Huang, J. A. dos Santos, S. Tubaro, and A. Rocha, "Data-driven feature characterization techniques for laser printer attribution," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 8, pp. 1860–1873, Aug. 2017.

[52] S. Joshi, M. Lomba, V. Goyal, and N. Khanna, "Augmented data and improved noise residual-based CNN for printer source identification," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 2002–2006.

[53] M. Bibi, A. Hamid, M. Moetesum, and I. Siddiqi, "Document forgery detection using printer source identification—A text-independent approach," in *Proc. Int. Conf. Document Anal. Recognit. Workshops (ICDARW)*, Sep. 2019, pp. 7–12.

[54] J. Bromley, J. W. Bentz, L. Bottou, I. Guyon, Y. Lecun, C. Moore, E. Säckinger, and R. Shah, "Signature verification using a 'Siamese' time delay neural network," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 7, pp. 25–44, Aug. 1994.

[55] B. Li, W. Wei, A. Ferreira, and S. Tan, "ReST-Net: Diverse activation modules and parallel subnets-based CNN for spatial image steganalysis," *IEEE Signal Process. Lett.*, vol. 25, no. 5, pp. 650–654, May 2018.

[56] J. Fridrich and J. Kodovsky, "Rich models for steganalysis of digital images," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 3, pp. 868–882, Jun. 2012.

[57] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[58] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1026–1034.

[59] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.

[60] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1800–1807.

[61] M. D. Zeiler, "ADADELTA: An adaptive learning rate method," 2012, *arXiv:1212.5701*.

[62] J. C. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *J. Mach. Learn. Res.*, vol. 12, no. 61, pp. 2121–2159, 2011.

[63] T. Dozat, "Incorporating Nesterov momentum into Adam," in *Proc. 4th Int. Conf. Learn. Represent.*, 2016, pp. 1–4.

[64] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015, *arXiv:1512.03385*.

[65] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," 2019, *arXiv:1905.11946*.

[66] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," 2016, *arXiv:1608.06993*.

[67] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," 2018, *arXiv:1801.04381*.

[68] A. Ferreira and M. Barni, "Attacking and defending printer source attribution classifiers in the physical domain," in *Proc. 2nd Workshop MultiMedia FORensics WILD*, 2023, pp. 347–363.

[69] N. Purnekar, B. Tondi, and M. Barni, "Physical domain adversarial attacks against source printer image attribution," in *Proc. Asia–Pacific Signal Inf. Process. Assoc. Annu. Summit Conf. (APSIPA ASC)*, Macau, Dec. 2024, pp. 1–6.

NISCHAY PURNEKAR (Student Member, IEEE) received the M.Sc. degree in electronics and telecommunication engineering from the University of Siena, Siena, Italy, where he is currently pursuing the Ph.D. degree with the Department of Information Engineering and Mathematics. His research interests include adversarial machine learning, with a particular focus on physical-domain adversarial attacks, the robustness of computer vision systems, and multimedia security. He has worked on adversarial robustness for license plate detection, deep learning-based authentication of printed patterns, and the security of visual recognition systems. His broader interests encompass adversarial attack and defense strategies, robust feature learning, and secure deep learning applications in real-world scenarios.

GIACOMO CANCELLI received the Ph.D. degree in computer engineering from the University of Siena, in 2009, focusing on data hiding techniques. Since 2010, he has been an entrepreneur with a strong interest in emerging technologies and innovation. His work spans the development of anti-counterfeiting technologies, artificial intelligence applications, and the integration of advanced AI techniques into highly innovative commercial products. He is the inventor of several patents in the field of anti-counterfeiting and has contributed to the application of AI in product development. He collaborates with the University of Siena on research projects related to image processing, signal encoding and compression techniques, and compiler and parser development systems. His research interests include steganalysis, steganography, digital watermarking, and the intersection of AI, cybersecurity, and data analysis.

ANSELMO FERREIRA (Member, IEEE) received the Ph.D. degree in computer science (Hons.) from the State University of Campinas, Brazil, in 2016.

Since then, he acted as a Postdoctoral Fellow in several research institutes and universities in Brazil, China, and Italy, researching and developing machine learning solutions focused on industry, forensics, and market applications. He currently holds a Marie Skowłodska Curie Postdoctoral Fellowship at the University of Siena, Siena, Italy, in European Union Project PrintOut, researching and developing machine learning and computer vision solutions for printed document forensics and package anticounterfeiting. His research interests include computer vision, multimedia forensics, and big data analysis. He is also an Elected Member of the IEEE Information Forensics Technical Committee, where he is part of the technical directions subcommittee, where he acts to identify future trends and directions in information forensics. He has also been working as a reviewer of dozens of conferences and journals.

MAURO BARNI (Fellow, IEEE) received the Ph.D. degree in informatics and telecommunications from the University of Florence, in 1995. During the last three decades he has been studying the application of image processing techniques to copyright protection and authentication of multimedia, and the possibility of processing encrypted signals without decrypting them. Lately, he has been working on theoretical and practical aspects of adversarial signal processing and security of machine learning. He is the author of about 350 articles and holds five patents in digital watermarking and image authentication. He is a fellow of EURASIP. He was a recipient of the Individual Technical Achievement Award of EURASIP (2016). He has been the Chairperson of the IEEE Information Forensic and Security Technical Committee (2010–2011). He was the Technical Program Chair of ICASSP 2014. He was appointed DL of the IEEE SPS from 2013 to 2014. He has been the Editor-in-Chief of IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY (2015–2017). He was the Founding Editor of the *EURASIP Journal on Information Security*. He has been serving as Associate Editor of many journals, including several IEEE TRANSACTIONS.