

Data-Driven Feature Characterization Techniques for Laser Printer Attribution

Anselmo Ferreira, Luca Bondi, *Student Member, IEEE*, Luca Baroffio, Paolo Bestagini, *Member, IEEE*, Jiwu Huang, *Fellow, IEEE*, Jefersson A. dos Santos, Stefano Tubaro, *Senior Member, IEEE*, and Anderson Rocha, *Senior Member, IEEE*

Abstract—Laser printer attribution is an increasing problem with several applications, such as pointing out the ownership of crime proofs and authentication of printed documents. However, as commonly proposed methods for this task are based on custom-tailored features, they are limited by modeling assumptions about printing artifacts. In this work, we explore solutions able to learn discriminant-printing patterns directly from the available data during an investigation, without any further feature engineering, proposing the first approach based on deep learning to laser printer attribution. This allows us to avoid any prior assumption about printing artifacts that characterize each printer, thus highlighting almost invisible and difficult printer footprints generated during the printing process. The proposed approach merges, in a synergistic fashion, Convolutional Neural Networks (CNNs) applied on multiple representations of multiple data. Multiple representations, generated through different pre-processing operations, enable the use of small and lightweight CNNs whilst the use of multiple data enable the use of aggregation procedures to better determine the provenance of a document. Experimental results show that the proposed method is robust to noisy data and outperforms existing counterparts in the literature for this problem.

Index Terms—Laser printer attribution; deep learning-based document provenance analysis; convolutional neural networks; multiple representation; multiple data.

I. INTRODUCTION

Printed documents are found everywhere. From simple documents available today such as homeworks and warnings, to more crucial ones such as contractual clauses and scientific articles, a printer is always involved, being it a dot matrix, dye-sublimation, thermal, ink-jet or laser. The last one has been the choice of ordinary people and offices in the last decade because of its speed, quality of printing and decreasing price.

However, with this massive access to printing devices, a new threat has also emerged: the use of laser printers for criminal intentions. Additional contractual clauses inexistent before, child pornography and animal abuse photos, life threatening letters, illegal correspondence, terrorist plots, fake currency

and fake documents can now be easily printed by anyone. Hence, providing ways of pinpointing printing ownership of documents is paramount, mainly to link them to criminals. Also, linking a document to a printer is another way of authenticating official documents.

Several approaches have been proposed for this task in the literature. Some techniques are based on laboratory analysis of the actual used paper [1], [2]. However, these methods can damage or even destroy investigated documents as chemical and physical procedures are involved. Another branch of approaches exploits the so called extrinsic signatures, which are characteristic footprints printed on documents, either visible or not to the naked eye. These signatures can be embedded into printed material by modifying the printing process to encode some sort of source identification [3]. This can be done, for example, using watermarks, pulse width modulation, QR-codes or security deterrents [4], [5], [6], [7]. Recently, it has been reported that some printers encode, on the printed pages, some provenance information using tiny yellow dots spread over the printing material, no matter if the document is colored or not [8], [9]. The limitation of these approaches is the fact that they do not represent a golden standard followed by the whole industry, and an expert user can change the printer's firmware maliciously.

Finally, another group of methods aims at solving printer attribution in a non-invasive (i.e., preserving the original document) blind fashion. This means these methods do not rely on printer information embedded into documents. Rather, they rely on signatures left by mechanical imperfections specific of printers that can be searched for on the printed material [10], [11], [12], [13]. These techniques use computer vision and machine learning approaches applied to scanned versions of suspected documents. More specifically, existing methods for text (non-colored) documents make use of hand-crafted features generated by an initial assumption about printing imperfections. These features are then extracted from a limited part of the data (e.g., one symbol or letter of the raw text) [11], [14], [15], [16] and fed to supervised classifiers for reaching a decision upon the printer source of the document.

As the use of engineered features has been recently challenged by feature learning paradigms in many detection and recognition tasks [17], in this paper, we present a data-driven printer attribution approach. This is the first deep learning solution for laser printer attribution that uses several Convolutional Neural Networks (CNNs) in parallel, extracting meaningful discriminative patterns straight from the analyzed documents

Anselmo Ferreira and Jiwu Huang are with the Shenzhen Key Laboratory of Media Security, College of Information Engineering, Shenzhen University, Shenzhen, P. R. China

Anderson Rocha is with the Institute of Computing - University of Campinas (Unicamp) - Av. Albert Einstein, 1251, CEP 13083-852, Campinas, São Paulo, Brazil.

Jefersson A. dos Santos is with the Department of Computer Science - Universidade Federal de Minas Gerais - Av. Antônio Carlos 6627 - Prédio do ICEx - Pampulha, Belo Horizonte-MG, Brazil, 31270-010.

Luca Bondi, Luca Baroffio, Paolo Bestagini and Stefano Tubaro are with the Dipartimento di Elettronica Informazione e Bioingegneria - Politecnico di Milano - Piazza Leonardo da Vinci 32, 20133, Milano, Italy.

instead of using ordinary feature engineering. Our approach exploits the advantages of back-propagation procedures, commonly used in CNNs, to automatically learn discriminant features from a set of existing training documents. It also uses different data representations to better identify printing pattern artifacts on an input printed character, further enhancing the characterization process and the analysis of provenance of a printed document (attribution task). Finally, we apply a late-fusion paradigm to integrate the classification outcomes coming from different letters within the same document.

The proposed approach is tailored to blind laser printer attribution for grayscale text documents. This means we do not rely upon any prior information such as inserted watermarks, and the traces we exploit can be extracted by the analysis on sets of letters. As the proposed method builds upon a machine learning framework, we assume the availability of a set of training documents as for any other supervised learning approach in the literature [12], [18]. More specifically, we consider that the only available data are scanned versions of: (i) the questioned document, and; (ii) a set of training documents coming from a set of suspect or candidate printers. The available training documents are considered to be printed with the same font and approximately the same font-size of the document under analysis. Moreover, we assume that some training documents actually come from the printer used to generate the document under investigation. In this setup, we consider that all the documents have been scanned with the same scanner, in order to avoid introducing any additional bias.

Notice that, even though these hypotheses may seem strict, we are not bounding our method neither to work with a single font and font-size, nor to work with a fixed character. Moreover, in courts of law, it is common that: (i) the analyst has direct access to many documents printed with the suspect printer, or; (ii) the analyst has access to the suspect printer itself. In the first case, as the commonly used fonts and sizes for official documents are not many, the analyst has a high probability of owning sufficient data with the same font and (approximate) size. In the second case, the analysis is even simpler, as the analyst can print as many documents he/she wants, with any font and size.

In summary, the main contributions of this paper are:

- 1) The design and development of an ad-hoc CNN for laser printer attribution based on the analysis of small patches representing text characters. The network is characterized by a small amount of parameters, thus allowing a fast yet reliable training with a limited set of labeled data.
- 2) The use of CNNs on multiple representations of the same character to learn complementary features that are fused together for an increased recognition accuracy.
- 3) The use of a late-fusion paradigm to merge results coming from the analysis of different characters within the same document. In this way, each character is classified separately, and individual results contribute to the final document label. This is useful especially for documents containing repetitions of some letters.

We organized the remaining of this paper into six sections. Sec. II discusses the most important techniques in the liter-

ature to identify the intrinsic artifacts of laser printers using computer vision and machine learning approaches. Sec. III introduces the concept of CNN, which is necessary for understanding the rest of the work. Sec. IV introduces our approach for source printer attribution. Sec. V reports all the details about the experimental methodology used for validating the proposed method and compare it to the existing counterparts in the literature while Sec. VI shows the performed experiments and results. Finally, Sec. VII concludes this work reporting our final considerations and proposals for future work.

II. LITERATURE SOLUTIONS FOR LASER PRINTER ATTRIBUTION

Laser Printers (LPs), differently from ink-jet printers, use a dry painting process based on the electromagnetic attraction of sooty powders inside a toner and the paper to be printed, in a process conducted by modifying charges on a light-sensitive revolving drum by a laser light source reflected by mirrors. The laser printer process occurs, in a nutshell, by charging this drum by a laser reflected by a mirror, which attracts the positive charged toner. Finally, the paper attracts the toner and a fusing process, by heat, joins the toner to the paper.

The intrinsic characteristics that can be seen on printed pages during this process are generated by imperfections in the manufactured parts of LPs, such as the leak of electric charges in some parts of the drum, different patterns of mirrors angle for different manufacturers, different speed of the revolving drum, among others. One of these intrinsic characteristics is called banding and is the most considered by the literature. Banding is defined as light and dark lines in a perpendicular direction to where the paper is moved inside the printer [4], [19]. Different brands are characterized by almost unique banding frequencies on different models of printers [12]. Several techniques in the literature have been focused on detecting such banding artifacts. Most of them can be divided in approaches focused on color documents (images) and text-only-documents. We discuss both of them in the following subsections.

A. Solutions for color documents

Existing methods to identify the source printer of color documents (i.e., documents with images) often exploit intrinsic signatures in the printing process, such as noise and geometric distortions, or in statistics derived from the transformed scanned images.

1) *Solutions based on noise analysis:* Lee et al. [20], [21] used the CMYK color space to detect the printer source of a document. The authors calculate a residual image by subtracting the scanned version of a document to its Wiener-filtered version. The residual image is then summarized using gray-level co-occurrence matrix (GLCM) statistics [22] and classified using a machine learning algorithm. Following a similar path, Choi et al. [23] and Tsai et al. [24] incorporated different color channels in the analysis and employed wavelets for feature extraction.

Elkasrawi and Shafait [18] also used the noise residual pattern to identify the printer even with common-resolution

scans (400dpi). For this they propose a descriptor based on the work of Khanna et al. [25], in which statistics of the row and column directions of the image are calculated. However, image filtering is performed differently, with the aid of the Otsu's threshold method [26].

2) *Solutions based on the analysis of geometric distortions:* Bulan et al. [27] used geometric distortions to identify the source of a color document. First, geometric signatures are extracted by estimating the positions of dots in halftone in training scanned documents of a given set of printers. Then, by correlation, the halftone points in a test document are linked to their source. Wu et al. [28] created printer models composed of distance and angles of halftone dots. K-means clustering on these Euclidean distances help in the final printer attribution process.

3) *Solutions based on the analysis of statistics of the transformed image:* Ryu et al. [29] proposed the analysis of very high-resolution scanned images through histograms of Hough transform angles in CMYK color channels, generating a feature vector of printing patterns for each document printed by a given printer. The printer attribution is performed by correlating this pattern with a reference created for each printer.

Kim and Lee [30] used the halftone patterns for laser printer identification, acquiring images by photography, instead of scanning. First, the image is preprocessed to eliminate illumination variability using each channel in the CMY domain. Then, a set of 15 halftone texture features are extracted in the discrete Fourier transform domain and are used to feed a machine learning classifier. This work was extended upon in [13] using the curvelet transform and correlation-based attribution.

B. Solutions for text documents

For text documents, most of the approaches to printer attribution rely upon texture, noise and geometric distortion analysis in the printed letters to find the extrinsic signatures of the banding process common to different printers.

1) *Solutions based on the texture of printed letters:* Mikkilineni et al. [10], [31] proposed the use of texture descriptors based on statistics of gray-level co-occurrence matrices to identify the source of text documents. A set of letters "e", which is the most used letter in English texts [32], is chosen for feature extraction. Then, 22 statistics of gray-level co-occurrence matrices are extracted and used as input to a previously trained 5-nearest neighbors classifier, with the majority voting of the classified letters defining the final source of a document. In follow-up works, support vector machines (SVM) were used [14], as well as clustering and Euclidean distances [33]. Jiang et al. [34] proposed the extraction of feature vectors based on Benford's law. The extracted features were the first digit probability distribution of discrete cosine transform coefficients from multi-size blocks. Following a different path, Ali et al. [35] used the linearized pixel values of letters "i" as features properly mapped onto lower dimensional spaces through Principal Component Analysis. The decision making is then performed using a Gaussian mixture model machine learning classifier.

Ferreira et al. [12] proposed a series of approaches based on the multidirectionality and multiple resolution banding texture effects present in printed letters in a document. The authors extended the GLCM texture descriptor to consider more directions and scales in the analysis of the input letter. They also proposed another descriptor, called the convolutional texture gradient filter, which filters textures with specific gradient, present in areas that better differentiate the printers. The authors used the proposed approaches on "e" letters and proposed to consider another region for analysis: the frames, which are rectangular areas with sufficient printing material.

Finally, other authors have focused on the attribution problem for languages using different alphabets. Tsai et al. [36], [15] combined features from statistics of gray level co-occurrence matrices and sub-bands of wavelet transform for laser printer source of Chinese printed documents. As with English language, a specific symbol of Chinese language was chosen for analysis. Tsai et al. [16] extended upon this method by using statistical features from a gray-level co-occurrence matrix, discrete wavelet transform, spatial filter, Wiener filter and Gabor filter to identify the source of Japanese printed documents.

2) *Solutions based on the analysis of noise and geometric distortions:* Kee and Farid [11] proposed to use reference characters and the reconstruction error to identify the source of text documents. The authors start with a reference "e" character of each printer. Then the search of similar characters from the same printer is done in a training step by template matching. These letters are then used to build the printer profile, useful for printer attribution later on. This profile is firstly built by preprocessing letters with histogram normalization and registration with respect to the reference letter of the printer. Then the mean character is calculated and the top p eigenvectors from principal component analysis [37] are calculated on aligned characters, yielding the printer profile.

Wu et al. [38] used geometric distortions to identify the laser printer source of documents. They first model a projective transformation using the center of characters and the whole scanned image in uncompressed format. Then, they solve this model with least squares and singular value decomposition for outliers removal. The estimated model parameters are used as geometric signatures inserted in a machine learning classifier. Finally, Schreyer [39] used statistical features from noise images in the discrete cosine transformed domain and in the multi-resolution wavelet domain, to train a machine learning classifiers for source printer attribution.

C. Remarks

In this work, instead of grappling with the printer attribution problem with hand-crafted features, similarly to previous solutions in the literature, we set ourselves the following guiding research principles:

- 1) Learn the discriminative features directly from the available collected data in a totally data-driven fashion.
- 2) Extract meaningful discriminative characteristics from a reduced set of training data, instead of the large ones

often necessary for training deep Convolutional Neural Networks.

Before we discuss how we deal with these requirements in our proposed method to perform laser printer attribution, it is worth discussing some basic concepts about deep neural networks. We do this in the next section.

III. CONVOLUTIONAL NEURAL NETWORKS

Since 2012 [40], convolutional neural networks have shown to be very effective in complex image classification tasks. Pioneered by LeCun et al. [41], the main benefit of using CNNs with respect to traditional fully-connected neural networks is the reduced amount of parameters to be learned. Convolutional layers made of small size kernels allow an effective way of extracting high-level features that are fed to fully-connected layers. The training of a CNN is performed through back-propagation and stochastic gradient descent. The misclassification error drives the weights update of both convolutional and fully-connected layers. After training, it is possible to use the output of a network's layer as feature vector paired with an external classifier, rather than simply relying on network classification layer. The basic layers of a CNN are listed below:

- 1) **Input layer:** where data is fed to the network. Input data can be either raw image pixels or their transformations, whichever better emphasize some specific aspects of the image.
- 2) **Convolutional layers:** contain a series of filters with fixed size used to perform convolution on the image data, generating a so called feature map. These filters can highlight some patterns helpful for image characterization, such as edges, regular patterns, etc.
- 3) **Pooling layers:** these layers ensure that the network focuses only on the most important patterns yielded by convolution and ReLU. A pooling layer summarizes the data by sliding a window across the feature maps and applying some linear or non-linear operations on the data within the window, such as generalized mean or max, reducing the dimensionality of the feature maps used by the following layers.
- 4) **Rectified Linear Unit (ReLU):** ReLU layers normally follow a convolution operation and are responsible for applying a non-linear function to the output x of the previous layer, such as $f(x) = \max(0, x)$. According to Krizhevsky et al. [40], they can be used for fast convergence in the training of CNNs, speeding-up the training as they deal with the vanishing gradient problem by keeping the gradient more or less constant in all network layers.
- 5) **Fully-connected layers:** used for the understanding of patterns generated by the previous layers. They are located at the end of the network and act as classifiers, usually followed by a soft-max layer to determine the class associated to the input image.
- 6) **Soft-max layer:** typically used at the end of the network during training. It normalizes input values in order to guarantee they sum to one. In doing so, its output can

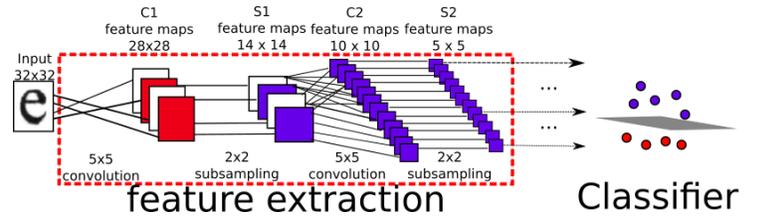


Fig. 1. Common architecture arrangement of a CNN. The input image is transformed into feature maps by the first convolution layer C1. A pooling stage S1 reduces the dimensions across the feature maps. The same process is repeated for layers C2 and S2. Finally, the last layer is used to classify the input.

be interpreted as a probability distribution (e.g., pointing out the probability of a sample to belong to each class).

Fig. 1 depicts one possible CNN architecture whose output at a given layer is fed to an external classifier. The type and arrangement of layers vary depending on the target application.

Although very powerful at representing patterns present in the data, the main drawback of deep learning is the fact that common CNNs normally need thousands or even millions of labeled data for training. This is an unfeasible condition in many applications due to the lack of training data and to the big amount of time needed to train a model. In this work, we present an alternative approach that deals with these requirements by considering several lightweight CNNs for laser printer attribution, as we shall discuss in Sec. IV.

IV. PROPOSED METHOD

The proposed solution for laser printer attribution works according to the following supervised machine learning pipeline. First, documents under analysis are digitalized and different sets of characters $\mathcal{S}_{\text{char}}$ are extracted from them (e.g., \mathcal{S}_e and \mathcal{S}_a for characters “e” and “a”, respectively). Each character of each set is processed separately. Characters are processed to obtain multiple representations of them (i.e., $\mathcal{S}_{\text{char}}^{\text{raw}}$, $\mathcal{S}_{\text{char}}^{\text{med}}$ and $\mathcal{S}_{\text{char}}^{\text{avg}}$ contains the raw, median filtered residual and average filtered residual versions of the characters). For each representation, different features $f_{\text{char}}^{\text{raw}}$, $f_{\text{char}}^{\text{med}}$ and $f_{\text{char}}^{\text{avg}}$ are extracted using small CNNs trained for this problem. These features are combined for each character set into a single feature vector f_{char} , which is used to classify each character separately. Finally, a voting step aggregates all labels $l_{\text{char}}^{\text{print}}$ assigned to each character into a final decision l^{print} . In the following, we provide a detailed description of each step.

A. Characters extraction

Choosing the appropriate input data to solve laser printer attribution problem with the proposed architecture is an important step. As a matter of fact, selected data should contain enough information to characterize the used printer (e.g., banding artifacts). However, this data should not be strongly influenced by the semantic of the content, otherwise the network training would be negatively affected. As a good compromise, and motivated by state-of-the-art methods using

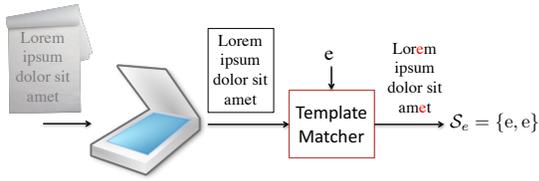


Fig. 2. Document digitalization and character extractor pipeline. Printed documents are scanned, and letters are extracted by template matching using the procedure described in [12]. The set S_e is composed by the detected pixel patches containing character “e”.

characters as the minimal entity for text documents analysis [12], we also decided to start the analysis at character level.

To extract characters from printed documents, a digital version of them must be obtained. To this purpose, our algorithm starts by scanning all documents under analysis and extracting from scanned versions sets of characters using the same extractor devised by Ferreira et al. [12], as shown in Fig. 2. The extractor works according to the following pipeline. First, we generate a reference letter, which has the same font typeface and is adjusted to have the same size as one letter from the scanned documents. Then, the algorithm slices the letter in eight regions and counts the proportion of black and white pixels in each one, yielding a feature vector used for letter extraction later on. To extract letters from the documents, black pixels connected components are extracted (i.e., character candidates) and the black/white ratio descriptor is computed again (the same as did before for the reference letter) for each connected component. Candidate letters whose descriptor has low cosine distance with respect to the reference letter descriptor are selected. Although the extractor is not perfect (the images of extracted letters have not the same size and some false positives may happen), it guarantees that most of the letters extracted are the same as the reference letter.

B. Multiple representation of input data

By using different characters and different representations of them, it is possible to separately train several small networks in parallel instead of a single complex network, thus reducing the computational complexity and still achieving promising results. The intuition is that: (i) several simpler deep networks can be effectively trained using less training examples and (ii) early layers of simple networks are sufficient to identify interesting artifacts contained in the pixel domain (e.g., banding). Moreover, we also decided to consider different representations of the input data along with multiple simple deep networks. Different data representations rather than raw image pixels have already been considered in the forensic literature, such as for median filtering detection [42].

To this purpose, from each document, different sets S_{char} of grayscale characters of the same font and approximately the same size are extracted. As an example, a set S_e of letters “e” and S_a of letters “a” are used. In order to exploit the advantages given by multiple representation, for each set S_{char} , we resorted to the following three different representations:

- 1) Raw data ($S_{\text{char}}^{\text{raw}}$): image pixels are used as input to the network as they are. This is the common representation



Fig. 3. Same letter “e” printed by different printers.



Fig. 4. Median filter residual representation of the same letters “e” showed in Fig. 3. Here, some minimal borders are highlighted. Pixel values (black and white) are inverted in this figure for better visualization.



Fig. 5. Average filter residual representation of the same letters “e” showed in Fig. 3. Here, natural borders are highlighted. Pixel values (black and white) are inverted in this figure for better visualization.

used as input for CNNs, as it contains high and low frequency components that can be isolated by the CNN filters and can be useful for image classification (see Fig. 3).

- 2) Median filter residual ($S_{\text{char}}^{\text{med}}$): we apply a 3×3 median filter over the image and subtract the image from the filtered version. The yielded noise pattern is used as input to the network. As the median filter better preserves edges, the median filter residual will contain, mostly, high frequency imperfections, which can be regarded as the banding (see Fig. 4).
- 3) Average filter residual ($S_{\text{char}}^{\text{avg}}$): we apply a 3×3 average filter over the image and subtract the image from its filtered version, using this residual as input to the network. This residual isolates border effects (see Fig. 5).

C. Feature extraction

To extract relevant features from our input data, we use a deep learning approach as discussed in Sec. III. More specifically, we train a simple CNN for each character and each set $S_{\text{char}}^{\text{raw}}$, $S_{\text{char}}^{\text{med}}$ and $S_{\text{char}}^{\text{avg}}$. Then we feed again patches from $S_{\text{char}}^{\text{raw}}$, $S_{\text{char}}^{\text{med}}$ and $S_{\text{char}}^{\text{avg}}$ to the networks to obtain three feature vectors $f_{\text{char}}^{\text{raw}}$, $f_{\text{char}}^{\text{med}}$ and $f_{\text{char}}^{\text{avg}}$ for each character within each set, using these vectors in a supervised classifier.

The used network architecture is common to each character and set and is similar in spirit to the MNIST network for digit recognition [43]. However, for a better representation of the data of interest herein, we train the network from scratch, yielding new filter weights able to recognize interesting characteristics for laser printer attribution. As far as we know, this is the first deep network custom-tailored to the printer attribution problem. The used CNN architecture has the following layers:

- 1) One input layer, where the raw image or a different representation (median filter residual or average filter residual) is used. It requires 28×28 images as input.
- 2) The first convolutional layer is made of 20 5×5 filters and is followed by a non-overlapping max pooling layer of size 2×2 and stride 2.

- 3) A second convolutional layer, with 50 filters of size $5 \times 5 \times 20$ is followed by another non-overlapping max pooling layer of size 2×2 and stride 2.
- 4) An inner product layer, which generates a vector $\in R^{500}$.
- 5) The 500 dimensional vector is non-linearly processed with a ReLU function applied element-wise.
- 6) An inner product layer acts as classifier with as many output confidence scores as the number of printers available during training.
- 7) A soft-max layer finally outputs the index and the confidence of the most probable printer.

In our proposed approach, we train the network using this architecture and then feed the training images again to the already trained the network, extracting 500-dimensional feature vectors in the last but one layer and repeating the process for the testing images. To follow the literature, we used the network as a feature extractor only, transferring the feature vectors to another and well used classifier for this application. The network autonomously learns which characteristics of the input images are relevant for discriminating the different printers.

Specifically, the network is trained using stochastic gradient descent with a momentum set to 0.9. We used an initial learning rate of 0.001 and a weight decay of 0.0005 without dropout. We used a batch size (subsampling of image examples used in one forward/backward pass through the network) of 100 images without batch normalization. The number of training epochs, which is the number of one forward and one backward pass of all training examples through the network was set to 30, and the model generated at the epoch with the smallest validation loss (20 epochs) was selected.

Fig. 6 and 7 depict the 20 5×5 filters of the first convolutional layer and also the characteristics they highlight from a letter printed by a given printer in the case the set $\mathcal{S}_e^{\text{raw}}$ is considered. These figures show that different filters enhance different areas of letters, such as texture and borders, which have been shown to be important to detect banding for LP attribution by existing methods in the literature such as [12].

D. Classification with early and late fusion

The proposed CNN architecture is characterized by a limited amount of parameters, in order to allow a fast and reliable

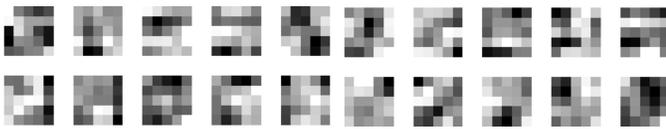


Fig. 6. Example of filters weights for the first convolutional layer operating on the raw input image pixels. Weight values are mapped in grayscale.

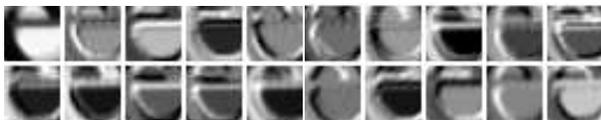


Fig. 7. Convolutional output of the first layer of the trained network, given an input letter from an investigated printer. For each filter, different areas inside or outside the borders are highlighted.

training even with a small number of labeled samples available. Small networks, as the one we are using, are expected to have worse performance with respect to bigger and deeper networks typically used in the computer vision community [17]. To compensate for this issue, we propose to use two lightweight fusion methods depicted in Fig 8:

- 1) **Early fusion – multiple representations of the same data:** we apply three different networks on input characters (of one type) coming from $\mathcal{S}_{\text{char}}^{\text{raw}}$, $\mathcal{S}_{\text{char}}^{\text{med}}$ and $\mathcal{S}_{\text{char}}^{\text{avg}}$. We concatenate the generated feature vectors $f_{\text{char}}^{\text{raw}}$, $f_{\text{char}}^{\text{med}}$ and $f_{\text{char}}^{\text{avg}}$ into a single vector f_{char} in an early-fusion fashion [44]. This vector is fed to a set of linear SVMs used with a One-vs-One classification policy [45] to classify each character separately assigning a label $l_{\text{char}}^{\text{print}}$ to each one of them. The rationale for using this technique is that different representations highlight complementary artifacts.
- 2) **Late fusion – multiple representations of different data:** after taking decisions at the character level within a document, we apply a late-fusion technique [44] by using majority voting on sets of different characters. This is useful especially when dealing with documents presenting a limited amount of characters within a single set (e.g., only a few “e” letters). The obtained document label l^{print} allows us to pinpoint which printer was used to print the document.

For final decision making, we analyze the list of classification outcomes (votes) of letters from a document. In the case of ties, we decide the mode as being the first most frequent value that appears in the list. For example, in a list

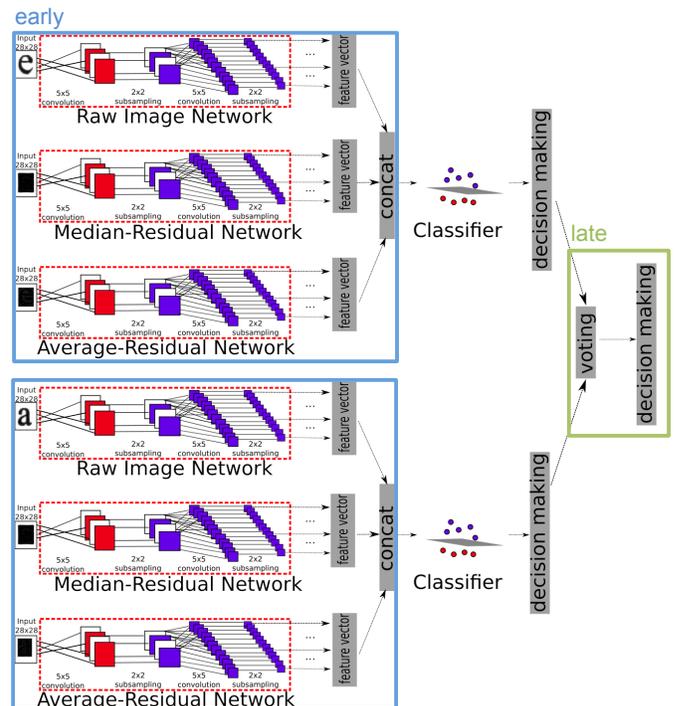


Fig. 8. Proposed multiple representations of different data for laser printer attribution through a set of lightweight Convolutional Neural Networks. Early and late fusion steps are highlighted in blue and green, respectively.

of classifications $x = [9, 7, 7, 7, 9, 9]$, the final classification would be 7. This can be thought of a pseudo-random tie-breaking and its most important advantage is simplicity. A more interesting tie-breaking policy would be summing up the distances to the hyperplane of each classified letter in a document, per class, and then deciding the final class as being the one with the highest sum (i.e., surer about the classification from the classifier).

E. Remarks

As we show later on in this paper, it is indeed possible to train effective deep learning networks (DNNs) with less data if we take appropriate actions such as: (i) not selecting a too deep network, (ii) learning the features on different, and complementary, representations of the data; and (iii) combining the different outputs in the end through fusion. That being said, our motivation for using a solution based on a DNN for feature extraction and a discriminative classifier at the end was threefold. First, we wanted to evaluate the richness of data-driven features directly and not the DNN as a full-fledged feature extractor + classifier. Although it is straightforward to attach a last soft-max layer in the end of the network for classification, we opted to use a discriminative classifier at the end to have a standardized form of comparison with previous works, which have used SVMs for classification. By doing this, we ended up having just one free comparison parameter (the features themselves). Second, our own previous experience with DNNs show that the combination of a DNN for feature extraction and a discriminative classifier at the end are very powerful, especially if we intend to perform fusion later on. Finally, our third motivation comes from that fact that by using a discriminative classifier at the end of the DNN-based feature extraction, we could simplify the fusion of different methods at the end, thus creating a lightweight integrated solution. The positive effects of these early and late fusion techniques will be discussed on Sec. VI.

V. EXPERIMENTAL SETUP

This section presents the experimental methodology used in this paper along with the used evaluation metrics, dataset and statistical tests. Finally, it details all the tested algorithms, some of which are baseline methods whereas some others are individual parts of our algorithm used to separately validate each step.

A. Dataset

For validation, we considered the same dataset of documents proposed by Ferreira et al. [12] and freely available for download at Figshare¹. It comprises 120 Wikipedia documents containing up to three pages each converted to Portable Document Format (PDF). These documents were printed by 10 printers using $75g/m^2$ letter paper and scanned using a 600 dpi resolution Plustek SO PL2546 device, generating a total of 1,184 images. Table I shows the printers breakdown

TABLE I
PRINTERS AND NUMBER OF DOCUMENTS PER PRINTER USED IN THE DATASET OF FERREIRA ET AL. [12]

ID	Brand	Model	Documents
B4070	Brother	HL-4070CDW	120
C1150	Canon	D1150	116
C3240	Canon	MF3240	120
C4370	Canon	MF4370DN	120
H1518	Hewlett Packard	CP1518	120
H225A	Hewlett Packard	CP2025	119
H225B	Hewlett Packard	CP2025	110
LE260	Lexmark	E260DN	119
OC330	OKI Data	C330DN	120
SC315	Samsung	CLP315	120
Total			1,184

along with their main characteristics. This is the first standardized dataset in the literature containing documents in two languages: English and Portuguese. Although the characters in these two languages appear to be similar, in Portuguese texts, there are some accentuation signals in some letters (e.g., é and ã) that can confuse the letter extraction or the classification.

In [12], the authors have used two different datasets, one considering regions of interest of 980×800 pixels extracted from the input documents — referred to as *Frames Dataset* — and another one with only detected and extracted characters from the input documents — referred to as *Letters/Characters Dataset*. After strongly reasoning about this problem, we further motivated our research to cope with the following real-world setups: (i) classifying documents for which only a few printed lines are available, making it impossible to extract many frames and end up with a reliable attribution solution in an investigation; and (ii) having available only small pieces of document, a torn apart document or a shredded one. Those cases would render the analysis of frames impossible or useless.

Based on this motivation, we set forth the objective of tailoring a solution to the problem that would allow us to have the highest possible attribution effectiveness while, at the same time, not requiring large input regions from the investigated document. Thus, we decided to use the *Letters/Characters Dataset* presented in [12] as our reference benchmark. In addition, we also established the objective of exploring data-driven features directly learned from the data instead of hand-crafted oriented solutions as the ones exploited and reported in [12]. For that, we would need inputs that would not lead to an explosion of parameters in our DNN-oriented solution.

In addition to only using the “e” letters extracted from the documents as [12], in this paper, we go beyond and exploit the impact of using different letters as well, as the authors in [12] did not consider these cases. Table II summarizes the datasets of letters used for the tests we generated from the aforementioned documents. As already mentioned, these have been extracted exploiting the characters extractor devised by Ferreira et al. [12]. With this method, we extracted several different letters of approximately 38×47 pixels printed with the Wikipedia font from the documents according to their frequency in the English language [32], resulting in four datasets of different letters $\mathcal{D}_e, \mathcal{D}_a, \mathcal{D}_d, \mathcal{D}_o$ as reported in Table II.

¹<http://dx.doi.org/10.6084/m9.figshare.1263501>

TABLE II
DATASETS USED FOR EXPERIMENTAL EVALUATION.

Dataset	Letter	Samples
\mathcal{D}_e	“e”	245,650
\mathcal{D}_a	“a”	286,098
\mathcal{D}_d	“d”	185,009
\mathcal{D}_o	“o”	351,850
$\tilde{\mathcal{D}}_e$	“e”	131,435
$\mathcal{D}_{\text{frame}}$	frames	352,433

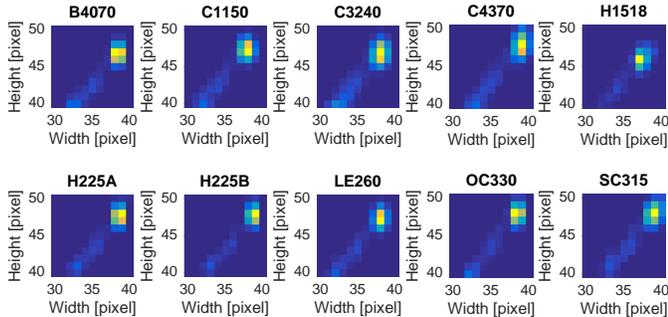


Fig. 9. Distribution of the extracted letter “e” sizes for each printer. Most of the characters have a resolution of 38×47 pixels, but some have slightly different sizes.

As the extractor in [12] also detects letters of similar font and size, these datasets can be regarded as affected by a small amount of noise. As an example, Fig. 9 shows the distribution of sizes of the extracted “e” letters for each printer. Although most of them share a common size, some of them slightly deviate. These datasets are then very useful to test the performance of the proposed algorithm in adverse conditions.

To validate the proposed method in a noiseless scenario, we also created a clean dataset $\tilde{\mathcal{D}}_e$ of 131,435 “e” letters. This dataset was created starting from the noisy “e” dataset \mathcal{D}_e keeping only the most similar letters sharing a $(38 \pm 1) \times (47 \pm 1)$ resolution.

At this point, it is worth mentioning that the input of our network is always a 28×28 pixel patch. Therefore, we always crop the center region of the letters so as to have inputs exactly matching this network requirement. We do not perform any resampling/resizing in order to avoid introducing additional processing artifacts that can hinder attribution performances by masking part of the telltales left behind by printers.

Finally, to validate our idea of using characters to train our small CNNs, we also built a dataset of small frames (e.g., small random patches). To this purpose, we applied a 28×28 frame extractor in the documents, extracting 300 valid frames whose ratio between black and white pixels r is $0.6 \leq r \leq 0.8$ from each scanned document. This resulted in the $\mathcal{D}_{\text{frame}}$ dataset.

B. Experimental methodology, evaluation metrics and statistical tests

For validation, we consider the same 5×2 cross-validation protocol used in [12]. In this protocol, we replicate the traditional 2-fold cross-validation protocol five times (thus 5×2). In each of these 2-fold cross validations, a set of documents (not characters) \mathcal{D} is split into \mathcal{D}_1 and \mathcal{D}_2 . In each of the five executions, a classifier is trained with characters

of documents in \mathcal{D}_1 and tested on characters present in \mathcal{D}_2 , and then vice-versa. After that, we report the results based on documents classification (after majority voting of test documents letters) and perform the statistical tests after 10 rounds of experiments. In this experimental protocol, each combination of training and test will use letters from 592 documents for training an one-against-one SVM classifier while the remaining 592 documents letters are used for testing the classifier. The number of letters used in the training and testing of each of 10 experiments (which we call fold) will depend on how many letters are extracted from each training and testing document and will also depend on which letter is being used in the analysis. For example, in the total 5×2 protocol, there are a mean of 122,825 letters ‘e’ for training and the same for testing. According to a study conducted by Dietterich et al. [46], the 5×2 cross-validation is considered an optimal experimental protocol for learning algorithms.

In a multi-class problem with n classes, the classification results may be represented in an $n \times n$ confusion matrix. In this case, the main diagonal contains the correct classifications while the other entries contain misclassifications. In the 5×2 cross validation protocol, one confusion matrix is yielded per experiment. Therefore, we present results by averaging these matrices.

To test the statistical relevance of the obtained experimental results, we consider a two-level statistical test. In the first level, we use the Friedman test as a pre-test to point out whether or not there is statistical difference in the obtained results. Then we refine these results with the Tukey-Kramer post-test, also known as honestly significant difference (HSD) test to point out statistical differences (if any) pairwise. In all tests, we set the confidence level to 95%.

C. Tested algorithms

We performed several tests to validate the proposed approach. First, we conducted a set of experiments aimed at selecting the reference CNN architecture. Then we tested each separate step of our algorithm (e.g., robustness to noise, early fusion, late fusion, etc.). Finally, we validated the proposed algorithm against state-of-the-art baseline methods.

At first we compared several different Convolutional Neural Network architectures in order to find the right balance between complexity and accuracy. To this purpose, in addition to the architecture proposed in Section IV, hereinafter denoted as \mathcal{S}^{2-Conv} , we also tested some deeper solutions. By adding one and two more convolutional layers, each followed by a max-pooling layer, we created two CNNs, denoted as \mathcal{S}^{3-Conv} and \mathcal{S}^{4-Conv} . Two additional state-of-the-art network architectures were used as benchmark: AlexNet [40], denoted as $\mathcal{S}^{AlexNet}$, and GoogLeNet [47], denoted as $\mathcal{S}^{GoogLeNet}$.

After validating the use of \mathcal{S}^{2-Conv} as CNN (hereinafter simply denoted as \mathcal{S} for the sake of clarity), we also tested each data representation separately. This means we extracted features using CNNs on a single representation of the input data (e.g., raw letters “e”) and used the obtained feature vectors for classification with SVM. Majority voting was applied to letters to take a decision at document level. As single

representations, we tested the median filter residual of the image ($\mathcal{S}_{\text{char}}^{\text{med}}$), the average filter residual ($\mathcal{S}_{\text{char}}^{\text{avg}}$) and the raw image pixels ($\mathcal{S}_{\text{char}}^{\text{raw}}$). We also tested different representations inspired by the existing methods in the literature. As a matter of fact, we tested the filtered image from the Convolutional Texture Gradient Filter (CTGF) using both the 3×3 ($\mathcal{S}_{\text{char}}^{\text{CTGF3}}$) and the 5×5 ($\mathcal{S}_{\text{char}}^{\text{CTGF5}}$) filters from the work of Ferreira et al. [12] and also the Wiener filter residual [48] ($\mathcal{S}_{\text{char}}^{\text{Wiener}}$). For each approach, the subscript “char” represents the letter we tested (e.g., $\mathcal{S}_{\text{e}}^{\text{Wiener}}$ for the Wiener-based representation on “e” letters). With an abuse of notation, we use the symbol \mathcal{S} to refer to both the algorithm and the set of input data.

We also tested the performance of the early fusion approach. For this, we concatenated the feature vectors from the last but one layer of CNNs applied on three different representations of the same data, making them the input of an SVM classifier. We refer to early fusion methods as $\{\mathcal{S}^{\text{raw}}, \mathcal{S}^{\text{med}}, \mathcal{S}^{\text{avg}}\}_{\text{char}}$, where the methods in the brackets represent the used data representations, and the subscript indicates the used letter (i.e., “e”, “a”, “d”, “o” or 28×28 frames).

To test the late fusion, we performed majority voting to classification labels obtained with early fusion methods run on different character families. We call these approaches $\{\mathcal{S}^{\text{raw}}, \mathcal{S}^{\text{med}}, \mathcal{S}^{\text{avg}}\}_{\text{char1}, \dots, \text{charN}}$, specifying the different sets of characters used for fusion. Notice that late fusion approaches also embed early fusion. The source-codes for all the proposed approaches will be available at GitHub² upon the acceptance of this paper. The same applies to all used dataset variations, which will be available through FigShare.

We also compared our proposed technique to eight state-of-the-art methods (see Sec. II) focused on text documents. The first one is the GLCM-based method from Mikkilineni et al. [10], [31], which describes the signature present in the banding with 22 statistics calculated per matrix. We refer to this approach in the experiments GLCM. The next four methods used in the experiments were proposed in the work of Ferreira et al. [12]. The first one uses GLCM with multiple directions (GLCM-MD), while the second uses GLCM with multiple directions and multiple scales in the input data (GLCM-MD-MS). The third one uses CTGF with size 3×3 (CTGF) and the fourth method uses a combination of all these methods (CTGF-GLCM-MD-MS).

The sixth implemented method from the literature was proposed by Kee and Farid [11] (RECONST-ERR) and uses reference characters to extract letters from documents. To detect the source of a document, letters “e” are extracted and compared with the profile of each printer to obtain a reconstruction error for each printer. The printer with the smallest mean error is detected as the source. Finally, we also tested two well-known texture descriptors widely used in the literature: (i) local binary patterns (LBP) [49]; and (ii) histogram of oriented gradients [50] (HOG).

VI. RESULTS AND DISCUSSION

We now turn our attention to the experimental results obtained with different methods. First, we test our proposed

lightweight CNN fusion approach against several individual CNN architectures. Second, we dissect the proposed approach to test each of its steps separately. Third, we show results considering the effects of training CNNs on noisy rather than noiseless data. Then, we compare different representations of the input data. Afterwards, we show the advantages of using multiple representations (early fusion) and multiple data (late fusion). Finally, we present experiments comparing the performance of our approach to the methods discussed in Sec. V-C. All experiments were performed using the methodology presented in Sec. V-B on the dataset with 1,184 printings presented in Sec. V-A.

A. Evaluation of the CNN model

The first step toward the development of our proposed deep learning approach for laser printer attribution is to determine the kind of CNN architecture that best suits the problem at hand. One natural solution would be using the whole digitalized document as input for a Convolutional Neural Network, but this procedure have the following drawbacks: (i) it requires the designing of deeper networks, which will require a larger amount of data, computational time and memory resources to train the network; and (ii) the network training process will be strongly influenced by the semantic of the documents. Conversely, smaller areas with fixed patterns used as input to smaller networks do not require as many layers as using the whole document as input and also can lead to a faster learning of network parameters and weights.

In this vein, we selected CNNs whose input are small patches of size as 28×28 , 227×227 and 224×224 as candidate architectures for our proposed multiple representation of multiple data approach. For each candidate architecture, we train and test the first split of the raw “e” dataset \mathcal{D}_{e} , training these architectures for 30 epochs. The model generated at the epoch with the smallest validation loss is selected as the best candidate for each CNN. We show in Table III results considering our fusion approach, denoted as $\{\mathcal{S}^{\text{raw}}, \mathcal{S}^{\text{med}}, \mathcal{S}^{\text{avg}}\}_{\text{a,e}}$, using six lightweight networks with 2 convolutional layers (architecture that we denote as $\mathcal{S}^{2\text{-Conv}}$) and some individual deeper architectures, using the networks as feature extractors and a linear SVM as the classifier.

TABLE III
RESULTS COMPARING DIFFERENT DEEP LEARNING APPROACHES FOR LASER PRINTER ATTRIBUTION IN ONE COMBINATION OF TRAINING AND TESTING. OUR BEST PROPOSED LATE FUSION APPROACH IS HIGHLIGHTED IN LIGHT GRAY. TTE REFERS TO THE TRAINING TIME FOR A SINGLE EPOCH.

Method	Accuracy	TTE [s]	Size [MB]	Input Data
$\{\mathcal{S}^{\text{raw}}, \mathcal{S}^{\text{med}}, \mathcal{S}^{\text{avg}}\}_{\text{a,e}}$	98.30%	20.22	9.84	$\mathcal{D}_{\text{a}}, \mathcal{D}_{\text{e}}$
$\mathcal{S}^{\text{GoogLeNet}}$ [47]	98.30%	886.00	39.40	\mathcal{D}_{e}
$\mathcal{S}^{\text{AlexNet}}$ [40]	98.13%	290.00	217.00	\mathcal{D}_{e}
$\mathcal{S}_{\text{e}}^{4\text{-Conv, raw}}$	97.29%	21.70	15.43	\mathcal{D}_{e}
$\mathcal{S}_{\text{e}}^{3\text{-Conv, raw}}$	96.10%	8.10	2.92	\mathcal{D}_{e}

As shown in Table III, the proposed approach, underpinned by six lightweight networks instead of one, has similar results

²<https://github.com/anselmoferreira/deep-learning-printer-attribution>

to a more complex network ($\mathcal{S}^{GoogLeNet}$) while presenting a memory footprint 75% more efficient and being, approximately, 43× faster to train. Moreover, each individual network of the proposed late-fusion approach consumes 1.64MB, thus the final footprint is $6 \times 1.64 = 9.84$ MB of space. This is a further confirmation that the use of the proposed lightweight simple networks in a fusion framework outperforms deeper solutions in terms of complexity-accuracy trade-off, at least for the particular setup considered herein. Indeed, the fusion approach with six networks reaches an accuracy equals the one generated by a deeper network, but with a reduced complexity.

We also evaluate the solutions for laser printer attribution with different training set sizes. We start comparing our proposed lightweight fusion of CNNs to existing solutions for laser printer attribution on different proportions of training data. For this experiment, we separated one combination of training and test data, sub-sampling the training data to be 1%, 10%, 30%, 50%, 70%, and 100% of the original training data, classifying the same testing data using the same SVM linear classifier used in the experiments. We show the results in Table IV. Each column shows a percentage of the training data used.

TABLE IV

RESULTS COMPARING DIFFERENT DEEP LEARNING APPROACHES AGAINST OUR PROPOSED APPROACH FOR LASER PRINTER ATTRIBUTION IN ONE COMBINATION OF TRAINING AND TESTING ON DIFFERENT AMOUNTS OF TRAINING DATA. THE BEST ACCURACY PER TRAINING DATA PROPORTION USED IS HIGHLIGHTED IN GRAY.

Method	1%	10%	30%	50%	70%	100%
$\{\mathcal{S}^{\text{raw}}, \mathcal{S}^{\text{med}}, \mathcal{S}^{\text{avg}}\}_{a,e}$	91.20%	97.29%	97.63%	97.63%	97.96%	98.30%
$\mathcal{S}^{GoogLeNet}$ [47]	87.64%	96.27%	94.92%	94.07%	98.13%	98.30%
$\mathcal{S}^{AlexNet}$ [40]	89.00%	96.44%	95.77%	96.44%	97.29%	98.13%

Normally, deeper networks require more data to show good results if compared to smaller ones. As Table IV shows, the proposed approach outperforms more complex CNNs for 1%, 10%, 30% and 50% of training data proportion used. A deeper network (GoogLeNet) starts to catch up and outperforms the proposed method when using 70% of data. In summary, the fusion of \mathcal{S}^{2-Conv} networks has the following advantages (i) it requires less data for effective training; and (ii) individually, each network (\mathcal{S}_2 architecture) used in the fusion requires less memory and time to train than using more complex networks. Therefore, we chose \mathcal{S}^{2-Conv} architecture in our proposed fusion approach. In the following, we will denote the \mathcal{S}^{2-Conv} architecture simply as \mathcal{S} in order to allow for a more compact notation.

B. Dealing with noisy data

In order to be useful in a real-world scenario, it is important that the developed method is robust against non-ideal working conditions. More specifically, it is paramount that the features learned by the CNNs are generalizable enough to guarantee good performance also on noisy data (e.g., letters of slightly different sizes). To test this property, we trained and tested the algorithm using different single representations of the “e” character (i.e., $\mathcal{S}_e^{\text{raw}}$, $\mathcal{S}_e^{\text{avg}}$, and $\mathcal{S}_e^{\text{med}}$) on different combinations of datasets (i.e., the noiseless $\tilde{\mathcal{D}}_e$ and the noisy \mathcal{D}_e).

Table V shows the achieved results. For each representation, the best accuracy (around 97%) is obtained when the algorithm is trained and tested on clean data not containing characters at different size ($\tilde{\mathcal{D}}_e$). When the same network trained on clean data ($\tilde{\mathcal{D}}_e$) is tested against dirty data (\mathcal{D}_e), accuracy falls down at approximately 85%. However, it is sufficient to train CNNs on \mathcal{D}_e to obtain results comparable to the noiseless case even when dirty data is tested (94%). Therefore, to ensure enough robustness, from this point on, we always consider noisy datasets for both training and testing, as they are closer to a real-world setup.

TABLE V
AVERAGE RESULTS USING EARLY FUSION AND SINGLE REPRESENTATIONS ON NOISELESS ($\tilde{\mathcal{D}}_e$) AND NOISY (\mathcal{D}_e) DATASETS.

Method	Mean	Training Data	Test Data
$\mathcal{S}_e^{\text{raw}}$	97.95%	$\tilde{\mathcal{D}}_e$	$\tilde{\mathcal{D}}_e$
	96.13%	\mathcal{D}_e	\mathcal{D}_e
	84.43%	$\tilde{\mathcal{D}}_e$	\mathcal{D}_e
$\mathcal{S}_e^{\text{avg}}$	97.56%	$\tilde{\mathcal{D}}_e$	$\tilde{\mathcal{D}}_e$
	94.50%	\mathcal{D}_e	\mathcal{D}_e
	85.81%	$\tilde{\mathcal{D}}_e$	\mathcal{D}_e
$\mathcal{S}_e^{\text{med}}$	96.87%	$\tilde{\mathcal{D}}_e$	$\tilde{\mathcal{D}}_e$
	94.30%	\mathcal{D}_e	\mathcal{D}_e
	85.58%	$\tilde{\mathcal{D}}_e$	\mathcal{D}_e

C. Choice of multiple representations

The proposed algorithm works exploiting multiple representations of the input data. It is therefore important to detect which representations contain more discriminative information for LP attribution. Table VI shows the best results obtained using different representations (e.g., $\mathcal{S}_{\text{char}}^{\text{raw}}$, $\mathcal{S}_{\text{char}}^{\text{Wiener}}$, etc.) on the different datasets (e.g., \mathcal{D}_a , \mathcal{D}_e , \mathcal{D}_o , etc.).

TABLE VI
RESULTS OBTAINED USING DIFFERENT REPRESENTATIONS ON DIFFERENT DATASETS SORTED FROM BEST TO WORST.

Method	Mean \pm Std.Dev.	Input Data
$\mathcal{S}_e^{\text{raw}}$	96.13% \pm 0.00	\mathcal{D}_e
$\mathcal{S}_a^{\text{avg}}$	94.89% \pm 0.30	\mathcal{D}_a
$\mathcal{S}_e^{\text{avg}}$	94.50% \pm 0.03	\mathcal{D}_e
$\mathcal{S}_e^{\text{med}}$	94.30% \pm 0.01	\mathcal{D}_e
$\mathcal{S}_a^{\text{med}}$	93.34% \pm 0.02	\mathcal{D}_a
$\mathcal{S}_a^{\text{raw}}$	93.07% \pm 0.03	\mathcal{D}_a
$\mathcal{S}_e^{\text{CTGF3}}$	89.12% \pm 0.03	\mathcal{D}_e
$\mathcal{S}_e^{\text{Wiener}}$	84.84% \pm 0.30	\mathcal{D}_e
$\mathcal{S}_e^{\text{CTGF5}}$	83.15% \pm 0.06	\mathcal{D}_e

Representations yielding higher accuracies are $\mathcal{S}_{\text{char}}^{\text{raw}}$, $\mathcal{S}_{\text{char}}^{\text{avg}}$ and $\mathcal{S}_{\text{char}}^{\text{med}}$, whereas the use of CTGF or Wiener-filtered versions of the characters provide the worst results. The best results are obtained using “a” and “e” datasets. This can be explained as they are the most common characters in English and Portuguese. Therefore, \mathcal{D}_a and \mathcal{D}_e are larger than \mathcal{D}_d , whereas \mathcal{D}_o probably is affected by too much noise as “o” can be often mistaken with other letters during the characters extraction phase.

Interestingly, for some data (letters), the raw representation in deep networks is not good enough. For instance, deep networks applied on average filter residual ($\mathcal{S}_a^{\text{avg}}$) of letters

“a” yielded an accuracy of 94.89%, against the accuracy of 93.07% on letters “a” raw image pixels (S_a^{raw}). This justifies the use of multiple representations and motivates the use of data fusion.

D. Early and late fusion

To validate the early and late fusion stages, we tested only the three selected best representations S_{char}^{raw} , S_{char}^{avg} and S_{char}^{med} . Table VII shows the results of the 5×2 cross-validation experiments considering this scenario.

TABLE VII
RESULTS COMPARING EARLY AND LATE FUSION USING THE BEST REPRESENTATIONS SORTED FROM BEST TO WORST. LATE FUSION APPROACHES ARE HIGHLIGHTED IN LIGHT GRAY.

Method	Mean \pm Std.Dev.	Input Data
$\{S_{a,e}^{raw}, S_{a,e}^{med}, S_{a,e}^{avg}\}$	97.33% \pm 0.00	D_a, D_e
$\{S_{a,e}^{raw}, S_{a,e}^{med}, S_{a,e}^{avg}\}_a$	96.89% \pm 0.00	D_a
$\{S_{a,e}^{raw}, S_{a,e}^{med}, S_{a,e}^{avg}\}_{a,e,d}$	96.87% \pm 0.00	D_a, D_e, D_d
$\{S_{a,e}^{raw}, S_{a,e}^{med}, S_{a,e}^{avg}\}_e$	96.84% \pm 0.00	D_e
$\{S_{a,e}^{raw}, S_{a,e}^{med}, S_{a,e}^{avg}\}_{a,e,o}$	96.24% \pm 0.03	D_a, D_e, D_o
$\{S_{a,e}^{raw}, S_{a,e}^{med}, S_{a,e}^{avg}\}_d$	93.67% \pm 0.03	D_d
$\{S_{a,e}^{raw}, S_{a,e}^{med}, S_{a,e}^{avg}\}_o$	92.21% \pm 0.03	D_o
$\{S_{a,e}^{raw}, S_{a,e}^{med}, S_{a,e}^{avg}\}_{a,e,frame}$	88.72% \pm 0.02	D_a, D_e, D_{frames}
$\{S_{a,e}^{raw}, S_{a,e}^{med}, S_{a,e}^{avg}\}_{frame}$	73.69% \pm 0.05	D_{frames}

Fusion approaches typically outperform the ones using only single representations. This is because different representations in the input layers of CNNs can contain important information that better identifies the banding over the different networks, as well as other printing artifacts left behind during the physical printing of a document. For example, banding in the borders contained in the average filter residual are better highlighted in its CNN and can complement the information found in the two other CNNs that use information from the raw image data and median filter residual. Moreover, different letters (late fusion) can contain even more explicit banding patterns than using the same letter. With these findings, we conclude that both multiple representation approach and late-fusion are useful for laser printer attribution using deep networks.

A special comment is in order regarding the use of frames ($\{S_{a,e}^{raw}, S_{a,e}^{med}, S_{a,e}^{avg}\}_{frame}$). As a matter of fact, their use, instead of letters, is not as effective when deploying a solution using deep learning. This is explained by the fact that different data are used as input at the same time to the same network, each of them presenting different printing patterns, probably demanding a different and deeper CNN architecture. This further confirms the idea of using characters for the proposed method.

Considering all the presented results, the statistical test using the Friedmann pre-test yielded the p-value of 7.55×10^{-154} , helping us to state that the approaches have a statistical significant difference. Table VIII shows the statistical Tukey HSD tests. This confirms that our proposed fusion approaches have statistically significant difference when compared to all the single representations. Notice that, even if the results obtained using early and late fusion are statistically equivalent, the use of late fusion is strongly motivated whenever a document does not contain enough letters from the same set (e.g., enough “e” letters).

TABLE VIII
TUKEY-HSD PAIRWISE STATISTICAL TESTS CONSIDERING CNN APPROACHES THAT USE UNIQUE AND MULTIPLE DATA.

Rank	Method	$\{S_{a,e}^{raw}, S_{a,e}^{med}, S_{a,e}^{avg}\}$	$\{S_{a,e}^{raw}, S_{a,e}^{med}, S_{a,e}^{avg}\}_a$	$\{S_{a,e}^{raw}, S_{a,e}^{med}, S_{a,e}^{avg}\}_{a,e,d}$	$\{S_{a,e}^{raw}, S_{a,e}^{med}, S_{a,e}^{avg}\}_e$	$\{S_{a,e}^{raw}, S_{a,e}^{med}, S_{a,e}^{avg}\}_{a,e,o}$	$\{S_{a,e}^{raw}, S_{a,e}^{med}, S_{a,e}^{avg}\}_d$	$\{S_{a,e}^{raw}, S_{a,e}^{med}, S_{a,e}^{avg}\}_o$	$\{S_{a,e}^{raw}, S_{a,e}^{med}, S_{a,e}^{avg}\}_{a,e,frame}$	$\{S_{a,e}^{raw}, S_{a,e}^{med}, S_{a,e}^{avg}\}_{frame}$	Total	
1	$\{S_{a,e}^{raw}, S_{a,e}^{med}, S_{a,e}^{avg}\}_{a,e}$	0	0	0	0	1	1	1	1	1	14	
2	$\{S_{a,e}^{raw}, S_{a,e}^{med}, S_{a,e}^{avg}\}_a$	0	0	0	0	1	1	1	1	1	12	
3	$\{S_{a,e}^{raw}, S_{a,e}^{med}, S_{a,e}^{avg}\}_{a,e,d}$	0	0	0	0	1	1	1	1	1	14	
4	$\{S_{a,e}^{raw}, S_{a,e}^{med}, S_{a,e}^{avg}\}_e$	0	0	0	0	0	1	0	1	1	11	
5	$\{S_{a,e}^{raw}, S_{a,e}^{med}, S_{a,e}^{avg}\}_{a,e,o}$	-1	0	-1	0	0	0	0	0	0	1	2
6	$S_{a,e}^{raw}$	-1	0	-1	0	0	0	0	0	0	1	2
7	$S_{a,e}^{med}$	-1	-1	-1	0	0	0	0	0	0	1	0
8	$S_{a,e}^{avg}$	-1	-1	-1	0	0	0	0	0	0	1	1
9	$S_{a,e}^{med}$	-1	-1	-1	0	0	0	0	0	0	1	0
10	$\{S_{a,e}^{raw}, S_{a,e}^{med}, S_{a,e}^{avg}\}_d$	-1	-1	-1	0	0	0	0	0	0	1	0
11	$S_{a,e}^{med}$	-1	-1	-1	0	0	0	0	0	0	1	0
12	$S_{a,e}^{raw}$	-1	-1	-1	0	0	0	0	0	0	1	0
13	$\{S_{a,e}^{raw}, S_{a,e}^{med}, S_{a,e}^{avg}\}_o$	-1	-1	-1	0	0	0	0	0	0	1	0
14	$S_{a,e}^{raw}$	-1	-1	-1	0	0	0	0	0	0	0	-14
15	$\{S_{a,e}^{raw}, S_{a,e}^{med}, S_{a,e}^{avg}\}_{a,e,frame}$	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0
16	$S_{a,e}^{med}$	-1	-1	-1	0	0	0	0	0	0	1	0
17	$S_{a,e}^{raw}$	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0
18	$\{S_{a,e}^{raw}, S_{a,e}^{med}, S_{a,e}^{avg}\}_{frame}$	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0

1 = Line method is better than column method
 0 = Line method is equivalent to column method
 -1 = Line method is worse than column method

E. Comparison with existing techniques in the literature

Table IX shows the results of the 5×2 cross-validation experiments considering our best approaches and existing counterparts in the literature. In this scenario, we are using all approaches as feature extractors and feeding a linear SVM classifier with these vectors in the training and testing step.

TABLE IX
RESULTS COMPARING THE BEST CONFIGURATIONS OF THE PROPOSED METHOD TO THE EXISTING METHODS IN THE LITERATURE AFTER 5×2 VALIDATION. LATE FUSION APPROACHES ARE HIGHLIGHTED IN LIGHT GRAY.

Method	Mean \pm Std.Dev.	Input Data
$\{S_{a,e}^{raw}, S_{a,e}^{med}, S_{a,e}^{avg}\}_{a,e}$	97.33% \pm 0.0065	D_a, D_e
$\{S_{a,e}^{raw}, S_{a,e}^{med}, S_{a,e}^{avg}\}_a$	96.89% \pm 0.0052	D_a
$\{S_{a,e}^{raw}, S_{a,e}^{med}, S_{a,e}^{avg}\}_{a,e,d}$	96.87% \pm 0.0087	D_a, D_e, D_d
$\{S_{a,e}^{raw}, S_{a,e}^{med}, S_{a,e}^{avg}\}_e$	96.84% \pm 0.0068	D_e
CTGF–GLCM–MD–MS [12]	96.26% \pm 0.0054	D_e
$S_e^{4-Com,raw}$	95.84% \pm 1.4700	D_e
$S_e^{3-Com,raw}$	95.40% \pm 0.8400	D_e
GLCM–MD–MS [12]	94.30% \pm 0.0110	D_e
GLCM–MD [12]	91.08% \pm 0.0089	D_e
HOG [50]	90.59% \pm 0.0214	D_e
LBP [49]	88.66% \pm 0.0145	D_e
RECONST–ERR [11]	78.90% \pm 0.0210	D_e
GLCM [10], [31]	77.87% \pm 0.0459	D_e
CTGF [12]	72.46% \pm 0.0377	D_e

Table IX shows that the first proposed method that outperforms the state-of-the-art is the one that uses multiple representations of the letter “e” ($\{S_{a,e}^{raw}, S_{a,e}^{med}, S_{a,e}^{avg}\}_e$), classifying, on average, three more documents in each fold of the cross validation when compared to the best existing solution in the

literature.

When using a different letter rather than “e”, such as the letter “a”, we also see an improvement in the results. The use of multiple representations of letter “a” ($\{S^{raw}, S^{med}, S^{avg}\}_a$) enables to classify a mean of four more documents in each fold when compared to state-of-the-art techniques. The multiple representation of multiple data “a” and “e” ($\{S^{raw}, S^{med}, S^{avg}\}_{a,e}$) shows its efficacy by showing the best overall accuracy of the experiments (97.33%), classifying six more documents than the best existing counterpart in the literature, on average. The reason for this good performance relies on the fact that this method takes into account multiple data with different banding artifacts that can be better highlighted using different representations in the specialized deep networks.

To validate the efficacy of the proposed methods, we also performed statistical tests. The Friedmann test showed a p-value of 3.16×10^{-138} , which helps us to state that the difference amongst the methods’ performance is statistically significant. Table X shows the Tukey-HSD pairwise tests.

TABLE X

TUKEY-HSD PAIRWISE STATISTICAL TEST RESULTS COMPARING THE PROPOSED METHODS TO THE EXISTING ONES IN THE LITERATURE.

Rank	Method	Attributed Printer													TOTAL	
		$\{S^{raw}, S^{med}, S^{avg}\}_{a,e}$	$\{S^{raw}, S^{med}, S^{avg}\}_a$	$\{S^{raw}, S^{med}, S^{avg}\}_{a,e,d}$	$\{S^{raw}, S^{med}, S^{avg}\}_e$	CTGF-GLCM-MD-MS [12]	S^4-Conv_{raw}	S^3-Conv_{raw}	GLCM-MD-MS [12]	GLCM-MD [12]	HOG [50]	LBP [49]	RECONST-ERR [11]	GLCM [10,32]		CTGF [12]
1	$\{S^{raw}, S^{med}, S^{avg}\}_{a,e}$	0	0	0	0	1	1	1	1	1	1	1	1	1	1	10
2	$\{S^{raw}, S^{med}, S^{avg}\}_a$	0	0	0	0	1	1	0	1	1	1	1	1	1	1	9
3	$\{S^{raw}, S^{med}, S^{avg}\}_{a,e,d}$	0	0	0	0	1	1	0	1	1	1	1	1	1	1	9
4	$\{S^{raw}, S^{med}, S^{avg}\}_e$	0	0	0	0	1	1	0	1	1	1	1	1	1	1	9
5	CTGF-GLCM-MD-MS [12]	-1	-1	-1	-1	0	0	0	0	1	1	1	1	1	1	2
6	S^4-Conv_{raw}	-1	-1	-1	-1	0	0	0	0	1	1	1	1	1	1	2
7	S^3-Conv_{raw}	-1	0	0	0	0	0	0	1	1	1	1	1	1	1	6
8	GLCM-MD-MS [12]	-1	-1	-1	-1	0	0	-1	0	0	0	1	1	1	1	-1
9	GLCM-MD [12]	-1	-1	-1	-1	-1	-1	-1	0	0	0	0	1	1	1	-4
10	HOG [50]	-1	-1	-1	-1	-1	-1	-1	0	0	0	0	1	1	0	-5
11	LBP [49]	-1	-1	-1	-1	-1	-1	-1	0	0	0	1	0	0	0	-7
12	RECONST-ERR [11]	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0	0	-11
13	GLCM [10,32]	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0	0	0
14	CTGF [12]	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0	0	0

1 = Line method is better than column method
 0 = Line method is equivalent to column method
 -1 = Line method is worse than column method

Considering the best performing configuration of our algorithm ($\{S^{raw}, S^{med}, S^{avg}\}_{a,e}$) and the best literature approach (CTGF-GLCM-MD-MS), Table XI and Table XII show confusion matrices representing the classification accuracies per printer. In Table XI, the confusion matrix of the proposed method shows that it is possible to identify 100% of three out of ten printers used in the experiments. These printers are Canon MF4370DN, OKI Data C330DN and Samsung CLP315. The CTGF-GLCM-MD-MS confusion matrix in Table XII, on the other hand, shows 100% classification for only one printer, the OKI Data C330DN.

It is also remarkable the fact that we are using two printers of the same model and brand (H225A and H225B) and it is possible to see, in Tables XI and XII, that there are just some

TABLE XI

CONFUSION MATRIX OF THE BEST PROPOSED APPROACH ($\{S^{raw}, S^{med}, S^{avg}\}_{a,e}$) SHOWING, IN PERCENTAGES, THE RIGHT AND WRONG MEAN HITS PER PRINTER AFTER THE 5×2 CROSS VALIDATION.

BEST PROPOSED	Attributed Printer									
	B4070	C1150	C3240	C4370	H1518	H225A	H225B	LE260	OC330	SC315
B4070	99.50									
C1150	0.52	99.48								
C3240	0.67		98.83	0.50						
C4370				100.00						
H1518	0.33	10.50			89.17					
H225A						93.10	6.90			
H225B						6.37	93.45			
LE260	0.18		0.17			0.33		99.50		
OC330									100.00	
SC315										100.00

TABLE XII

CONFUSION MATRIX OF THE BEST LITERATURE SOLUTION SHOWING, IN PERCENTAGES, THE RIGHT AND WRONG MEAN HITS PER PRINTER AFTER THE 5×2 CROSS VALIDATION.

CTGF-GLCM-MD-MS [12]	Attributed Printer									
	B4070	C1150	C3240	C4370	H1518	H225A	H225B	LE260	OC330	SC315
B4070	98.67	0.33	1.00							
C1150	1.72	98.28								
C3240			97.83	2.17						
C4370			1.00	0.50	98.50					
H1518	1.33	10.33			86.83	0.50			0.84	0.17
H225A					0.50		96.98	2.52		
H225B						12.90	87.10			
LE260						0.17		98.66		
OC330									100.00	
SC315										99.17

misclassifications between them. The errors in these cases are likely related to the printing artifacts generated by these two printers, which are similar for some documents. The proposed approach misclassified an average of 6.6% of the documents in these two classes, while the best existing method in the literature did it for 7.7% of the documents. It is also important to note that there are some misclassifications when classifying printers H1518 (an HP printer) and C3240 (a Canon printer) in both cases. This happens because these two printers present a slightly smaller average font size with respect to the other eight, as can be seen in Fig. 9. Therefore, they probably share some common artifacts.

VII. CONCLUSIONS AND FUTURE WORK

Laser printer attribution is a difficult task that involves investigating several printing patterns, created with different manufacturing processes, models and brands. Existing methods in the literature rely on computer vision and machine learning algorithms applied to scanned versions of documents, aiming at finding intrinsic signatures on printed material that better discriminate different printers. The main problem with these approaches is that they are underpinned by so-called hand-crafted features, which often require expert domain-knowledge to proper capture discriminative artifacts useful in the attribution process (e.g., intrinsic texture, geometric distortions in the printed material, etc.). Ideally, it would be interesting to also be able to detect important discriminative features directly from training data (data-driven methods). Those features could be even combined with hand-crafted ones for a more effective method.

In this vein, in this work, we have proposed a solution capable of learning discriminative features for the printer attribution problem directly from available training data (i.e., scanned versions of printed papers). The solution inherits the benefits of convolutional neural networks and back-propagation procedures, evolving the descriptor during training, thus making

these networks tailored to the analyzed data. The method relies on artifacts captured from different letters of documents in different languages. It also uses other letters rather than the commonly used “e”. To better highlight characteristic artifacts, different data representations through some image transformations were also investigated.

As we discussed thoroughly in this work, the use of multiple representations of multiple data allows to outperform the state of the art when for the laser printer attribution problem. Multiple representations fed as input to the used deep networks are important because they highlight different characteristics of the input images. We also showed that multiple representations of multiple data is a reasonable choice for laser printer attribution with deep networks. Indeed, with the benefits of the multiple representations presented before, multiple data also ensures a larger amount of voters per document.

One interesting finding in this research is the promising use of these different representations, composed by low-pass filtering residuals, as input to Convolutional Neural Networks. In a real-world setup, in which a suspect document was printed using a toner different from the one used for training the method, these low-pass filtering residuals can work better for pointing out the source than raw image inputs, as this last representation is more affected by the change of toner due to the increased presence of high-frequency components linked to toner artifacts. One natural extension of this proposed approach for this cross-dataset setup is replacing the raw image representation with other low-pass filtering residual analyses, such as the Gaussian filtering residual [51], bilateral filtering [52] and guided image filtering [53].

With current solutions to the printer attribution problem achieving high classification results, we believe it is time to aim at more daring challenges. For instance, current methods in the literature have shown great potential for classifying documents printed in similar conditions (both physical but also temporally close together. As a matter of fact, the printer attribution problem is much more difficult than its related problem of sensor attribution (for cameras and scanners). The reason is that the printing process has much more mechanical elements involved and intertwined when printing a document. Such elements surely play different roles in the creation of a unique signature for each printer. However, and the literature needs more study in this regard, it is natural that such signature will not last forever and will surely degrade over time as different elements in the printer age and defects appear. Then the next question is what happens if a document was printed several years ago and the printer under suspicion was just recently seized. A thorough investigation of this problem considering data captured in several moments along the years will be a significant contribution to the field.

Additional future work may be devoted to developing deep networks to be applied on different types of data, such as bigger letters and frames. Finally, we believe that also other different representations can be taken into account in further investigations. Finally, we also plan to study the behavior of the proposed approach on interpolated (rather than cropped) data.

ACKNOWLEDGMENT

This work was supported partly by NSFC (61332012, U1636202) and the Shenzhen R&D Program (JCYJ20160328144421330). We also thank the financial support of Brazilian National Council for Scientific and Technological Development (Grants #477662/2013-7, #304352/2012-8 and #449638/2014-6, and #304472/2015-8), the São Paulo Research Foundation – FAPESP (Grant #2015/19222-9, DéjàVu project), Minas Gerais Research Foundation – FAPEMIG (Grant APQ-00768-14), the Brazilian Coordination for the Improvement of Higher Level Education Personnel (Grant #99999.002341/2015-08 and the DeepEyes project) and Microsoft Research.

REFERENCES

- [1] A. Braz, M. Lopez-Lopez, and C. Garcia-Ruiz, “Raman spectroscopy for forensic analysis of inks in questioned documents,” *Forensic Science International*, vol. 232, no. 1-3, p. 206212, 2013.
- [2] P.-C. Chu, B. Y. Cai, Y. K. Tsoi, R. Yuen, K. S. Leung, and N.-H. Cheung, “Forensic analysis of laser printed ink by x-ray fluorescence and laser-excited plume fluorescence,” *Analytical Chemistry*, vol. 85, no. 9, pp. 4311–4315, 2013.
- [3] P.-J. Chiang, N. Khanna, A. Mikkilineni, M. Segovia, S. Suh, J. Allebach, G.-C. Chiu, and E. Delp, “Printer and scanner forensics,” *Signal Processing Magazine*, vol. 26, no. 2, pp. 72–83, 2009.
- [4] G. N. Ali, P.-J. Chiang, A. K. Mikkilineni, J. P. Allebach, G. T.-C. Chiu, and E. J. Delp, “Intrinsic and extrinsic signatures for information hiding and secure printing with electrophotographic devices,” in *International Conference on Digital Printing Technologies*, Sept 2003, pp. 511–515.
- [5] 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 Transactions on Information Forensics and Security (TIFS)*, vol. 11, no. 3, pp. 571–583, March 2016.
- [6] M. Gaubatz and S. Simske, “Printer-scanner identification via analysis of structured security deterrents,” in *Intl. Workshop on Information Forensics and Security (WIFS)*, 2009, pp. 151–155.
- [7] S. J. Simske, J. S. Aronoff, M. Sturgill, and J. C. Villa, “Spectral pre-compensation and security print deterrent authentication,” *NIP and Digital Fabrication Conference*, vol. 2008, no. 2, pp. 792–795, 2008.
- [8] E. F. Foundation, “Is your printer spying on you?” <https://www.fff.org/issues/printers>.
- [9] J. Frost, “The real reason you can’t print without colour ink: Printers leave tiny yellow dots for authorities to identify documents,” <http://www.techly.com.au/2015/09/24/real-reason-cant-print-without-colour-ink-printers-leave-tiny-yellow-dots-authorities-identify-documents/>, Sept 2015.
- [10] A. K. Mikkilineni, P. ju Chiang, G. N. Ali, G. T. c. Chiu, J. P. Allebach, and E. J. Delp, “Printer identification based on textural features,” in *International Conference on Digital Printing Technologies*, Oct 2004, pp. 306–311.
- [11] E. Kee and H. Farid, “Printer profiling for forensics and ballistics,” in *ACM Workshop on Multimedia and Security (MM&Sec)*, Sept 2008, pp. 3–10.
- [12] A. Ferreira, L. C. Navarro, G. Pinheiro, J. A. dos Santos, and A. Rocha, “Laser printer attribution: Exploring new features and beyond,” *Forensic Science International*, vol. 247, pp. 105 – 125, 2015.
- [13] D.-G. Kim and H.-K. Lee, “Colour laser printer identification using halftone texture fingerprint,” *Electronics Letters*, vol. 51, no. 13, pp. 981–983, 2015.
- [14] A. K. Mikkilineni, O. Arslan, P. ju Chiang, R. M. Kumontoy, J. P. Allebach, and G. T. c, “Printer forensics using SVM techniques,” in *International Conference on Digital Printing Technologies*, Oct 2005, pp. 223–226.
- [15] M.-J. Tsai, J.-S. Yin, I. Yuadi, and J. Liu, “Digital forensics of printed source identification for chinese characters,” *Multimedia Tools and Applications*, vol. 73, pp. 1–27, 2013.
- [16] M.-J. Tsai, C.-L. Hsu, J.-S. Yin, and I. Yuadi, “Japanese character based printed source identification,” in *IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2015, pp. 2800–2803.

- [17] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *International Conference on Computer Vision & Pattern Recognition (CVPR)*, June 2015, pp. 1–9.
- [18] S. Elkasrawi and F. Shafait, "Printer identification using supervised learning for document forgery detection," in *International Workshop on Document Analysis Systems*, April 2014, pp. 146–150.
- [19] N. Khanna, A. K. Mikkilineni, G. T. Chiu, J. P. Allebach, and E. J. Delp, "Survey of scanner and printer forensics at Purdue university," in *International Workshop on Computational Forensics (IWCF)*, Aug 2008, pp. 22–34.
- [20] H.-Y. Lee and J.-H. Choi, "Identifying color laser printer using noisy feature and support vector machine," in *International Conference on Ubiquitous Information Technologies and Applications*, 2010, pp. 1–6.
- [21] J.-H. Choi, H.-K. Lee, H.-Y. Lee, and Y.-H. Suh, "Color laser printer forensics with noise texture analysis," in *ACM Workshop on Multimedia and Security*, 2010, pp. 19–24.
- [22] R. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *Transactions on Systems, Man and Cybernetics*, vol. SMC-3, no. 6, pp. 610–621, 1973.
- [23] J.-H. Choi, D.-H. Im, H.-Y. Lee, J.-T. Oh, J.-H. Ryu, and H.-K. Lee, "Color laser printer identification by analyzing statistical features on discrete wavelet transform," in *Intl. Conference on Image Processing (ICIP)*, 2009, pp. 1505–1508.
- [24] M.-J. Tsai, J. Liu, C.-S. Wang, and C.-H. Chuang, "Source color laser printer identification using discrete wavelet transform and feature selection algorithms," in *IEEE International Symposium of Circuits and Systems (ISCAS)*, May 2011, pp. 2633–2636.
- [25] N. Khanna, G. T. C. Chiu, J. P. Allebach, and E. J. Delp, "Scanner identification with extension to forgery detection," in *SPIE Security, Forensics, Steganography, and Watermarking of Multimedia Contents*, March 2008, pp. 1–10.
- [26] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man and Cybernetics (TSMC)*, vol. 9, no. 1, pp. 62–66, Jan 1979.
- [27] O. Bulan, J. Mao, and G. Sharma, "Geometric distortion signatures for printer identification," in *Intl. Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2009, pp. 1401–1404.
- [28] H. Wu, X. Kong, and S. Shang, "A printer forensics method using halftone dot arrangement model," in *IEEE China Summit and International Conference on Signal and Information Processing (ChinaSIP)*, July 2015, pp. 861–865.
- [29] S.-J. Ryu, H.-Y. Lee, D.-H. Im, J.-H. Choi, and H.-K. Lee, "Electrophotographic printer identification by halftone texture analysis," in *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, March 2010, pp. 1846–1849.
- [30] D.-G. Kim and H.-K. Lee, "Color laser printer identification using photographed halftone images," in *European Signal Processing Conference (EUSIPCO)*, Sept 2014, pp. 795–799.
- [31] A. K. Mikkilineni, P. ju Chiang, G. N. Ali, G. T.-C. Chiu, J. P. Allebach, and E. J. Delp, "Printer identification based on graylevel co-occurrence features for security and forensic applications," in *SPIE Security, Steganography, and Watermarking of Multimedia Contents*, Mar 2005, pp. 430–440.
- [32] P. Micka, "Letter frequency (english)," <http://en.algoritmy.net/article/40379/Letter-frequency-English>.
- [33] A. K. Mikkilineni, N. Khanna, and E. J. Delp, "Forensic printer detection using intrinsic signatures," in *SPIE Media Watermarking, Security, and Forensics*, Feb 2011, pp. 1–11.
- [34] W. Jiang, A. T. S. Ho, H. Treharne, and Y. Q. Shi, "A novel multi-size block Benford's law scheme for printer identification," in *Pacific Rim Conference on Advances in Multimedia Information Processing*, 2010, pp. 643–652.
- [35] G. N. Ali, P. ju Chiang, A. K. Mikkilineni, G. T. Chiu, E. J. Delp, and J. P. Allebach, "Application of principal components analysis and gaussian mixture models to printer identification," in *International Conference on Digital Printing Technologies*, Oct 2004, pp. 301–305.
- [36] M.-J. Tsai and J. Liu, "Digital forensics for printed source identification," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2013, pp. 2347–2350.
- [37] R. Duda and P. Hart., *Pattern Classification and Scene Analysis*. John Wiley and Sons, 1973.
- [38] Y. Wu, X. Kong, X. You, and Y. Guo, "Printer forensics based on page document's geometric distortion," in *IEEE International Conference on Image Processing (ICIP)*, Nov 2009, pp. 2909–2912.
- [39] M. Schreyer, C. Schulze, A. Stahl, and W. Effelsberg, "Intelligent printing technique recognition and photocopy detection for forensic document examination," in *Informatiktag*, March 2009, pp. 39–42.
- [40] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Neural Information Processing Systems (NIPS)*, 2012, pp. 1106–1114.
- [41] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," in *Proceedings of the IEEE*, 1998, pp. 2278–2324.
- [42] J. Chen, X. Kang, Y. Liu, and Z. Wang, "Median filtering forensics based on convolutional neural networks," *IEEE Signal Processing Letters*, vol. 22, no. 11, pp. 1849–1853, Nov 2015.
- [43] VLFEAT, "MatConvNet: CNNs for MATLAB," <http://www.vlfeat.org/matconvnet/>, 2016.
- [44] C. G. M. Snoek, "Early versus late fusion in semantic video analysis," in *ACM International Conference on Multimedia (ACM-MM)*, Nov 2005, pp. 399–402.
- [45] A. Rocha and S. K. Goldenstein, "Multiclass from binary: Expanding one-versus-all, one-versus-one and ECOC-based approaches," *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)*, vol. 25, no. 2, pp. 289–302, Feb 2014.
- [46] T. G. Dietterich, "Approximate statistical tests for comparing supervised classification learning algorithms," *Neural Computation*, vol. 10, pp. 1895–1923, 1998.
- [47] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going Deeper with Convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1–9.
- [48] N. Wiener, *Extrapolation, Interpolation, and Smoothing of Stationary Time Series*. The MIT Press, 1964.
- [49] T. Ojala, M. Pietikainen, and D. Harwood, "A comparative study of texture measures with classification based on feature distributions," *Pattern Recognition*, vol. 29, no. 1, pp. 51–59, Jan 1996.
- [50] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *International Conference on Computer Vision & Pattern Recognition (CVPR)*, June 2005, pp. 886–893.
- [51] R. C. Gonzalez, R. E. Woods *et al.*, *Digital image processing*. Prentice hall Upper Saddle River, NJ, 2002.
- [52] F. Durand and J. Dorsey, "Fast bilateral filtering for the display of high-dynamic-range images," *ACM transactions on graphics (TOG)*, vol. 21, no. 3, pp. 257–266, 2002.
- [53] K. He, J. Sun, and X. Tang, "Guided image filtering," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 6, pp. 1397–1409, 2013.