International Journal of Electrical and Computer Engineering (IJECE)
Vol. 15, No. 1, February 2025, pp. 356∼364
ISSN: 2088-8708, DOI: 10.11591/ijece.v15i1.pp356-364

ReRNet: recursive neural network for enhanced image correction in print-cam watermarking

Said Boujerfaoui[1], Hassan Douzi[1], Rachid Harba[2], Khadija Gourrame[1,2]
[1]IRF-SIC Laboratory, Faculty of Sciences, Ibn Zohr University, Agadir, Morocco
[2]PRISME Laboratory, Polytech Orléans, Orléans University, Orléans, France

Article Info

Article history:

Received Jun 16, 2024
Revised Sep 3, 2024
Accepted Oct 1, 2024

Keywords:

Corner detection
Fourier transform
Geometric distortions
Image watermarking
Neural networks
Print-cam process

ABSTRACT

Robust image watermarking that can resist camera shooting has gained considerable attention in recent years due to the need to protect sensitive printed information from being captured and reproduced without authorization. Indeed, the evolution of smartphones has made identity watermarking a feasible and convenient process. However, this process also introduces challenges like perspective distortions, which can significantly impair the effectiveness of watermark detection on freehandedly digitized images. To meet this challenge, ResNet50-based ensemble of randomized neural networks (ReRNet), a recursive convolutional neural network-based correction method, is presented for the print-cam process, specifically applied to identity images. Therefore, this paper proposes an improved Fourier watermarking method based on ReRNet to rectify perspective distortions. Experimental results validate the robustness of the enhanced scheme and demonstrate its superiority over existing methods, especially in handling perspective distortions encountered in the print-cam process.

Corresponding Author:

Said Boujerfaoui
IRF-SIC Laboratory, Faculty of Sciences, Ibn Zohr University
BP 8106, Dakhla District, Agadir 80000, Morocco
Email: said.boujerfaoui@edu.uiz.ac.ma

1. INTRODUCTION

Today, data plays a fundamental pillar for industries and businesses, fortified by the ongoing surge in technological progress [1]. These advances have improved the efficiency of data transfer, but have also introduced challenges such as unauthorized data manipulation, affecting copyright protection and data integrity. As a result, industries are now faced with the imperative task of seeking real-time solutions for secure data processing. Since the 1990s, digital watermarking has emerged as an important research direction, particularly with the rise of smartphones, making watermarking algorithms viable for mobile systems to meet industrial security challenges [2].

Print-cam image watermarking [3] involves embedding a watermark into an image intended to be printed on a paper medium and then freehandedly captured using a smartphone camera. This uncontrolled operation introduces major challenges linked to perspective distortions and desynchronization problems [4], as freehand captures at varying angles can distort the watermark and introduce artifacts. These alterations complicate watermark detection, making it difficult or inaccurate [5]. The print-cam watermarking process is illustrated in Figure 1.

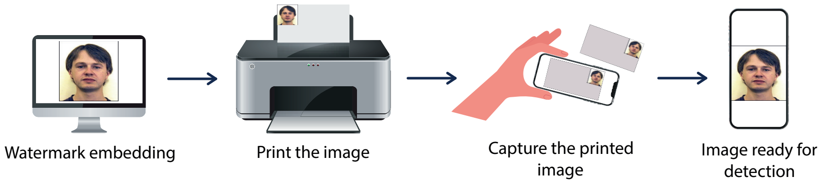


Figure 1. Print-cam watermarking process

Image watermarking approaches have advanced using spatial and frequency domains [6], [7] such as discrete cosine transform (DCT) [8], discrete wavelet transform (DWT) [9], and discrete Fourier transform (DFT) [10], each with distinct advantages and limitations. To deal with geometric distortions, various strategies have been devised, including frame synchronization mechanisms [11], convex optimization framework [12] and pseudo-random sequences [13]. Deep learning-based approaches, using convolutional neural networks, automate watermarking by learning correlations between watermarked and original images and exploiting imperceptible perturbations for data hiding [14]. However, research specific to print-cam scenarios remains limited, focusing mainly on learning-based techniques such as fine-tuned segmentation [15], [16], distortion mapping framework [17], [18], invariance layers [19], and 3D rendering distortion networks [20].

Image watermarking faces significant challenges in print-cam scenarios due to perspective distortions caused by the joint effect of rotation, translation, scaling (RST), and tilt angle. Despite their critical impact on watermark robustness, only a few approaches have been conducted for this process. Accordingly, leveraging deep learning methodologies and geometric transformation modeling could improve the robustness of watermarking schemes in overcoming these distortions.

This paper introduces randomized neural networks (ReRNet), a deep-learning-based method to address perspective distortions found in ID images during the print-cam process. ReRNet locates the ID image corners using a recursive convolutional neural network, enabling projective transformation for image rectification and accurate watermark alignment. As a result, an improved robust image watermarking technique is proposed to address print-cam attacks. This approach combines a Fourier transform-based embedding method [21] with ReRNet, employed for rectifying image distortions. The Fourier-based approach is selected for its proven ability to withstand the geometric attacks common in the print-cam process [22]. We conducted practical experiments on a selection of framed ID images, which were subjected to real print-cam attacks using a printer and two smartphones. The performance of the improved watermarking method was then evaluated and compared with existing competitive methods.

The rest of the paper is organized as follows: section 2 presents the complete watermarking method, including ReRNet architecture. Section 3 covers the experimental results. Finally, section 4 concludes the paper.

2. PRINT-CAM WATERMARKING SCHEME

First, the watermark is embedded into the original image. Once the watermarked image is printed and captured by a camera, perspective distortions are corrected using the proposed correction technique. Finally, in the detection phase, we determine whether or not the watermark is present in the corrected image. The components of the improved watermarking scheme are described below.

2.1. Fourier-based embedding

The watermark is embedded into the image DFT magnitude, specifically along a circular area with a defined radius r. The embedding process affects the luminance components of the original image, keeping the chrominance components unmodified. To improve the detection rate, a low-pass filter is employed on the embeddable DFT coefficients [21]. Hence, the watermark W is inserted into the filtered coefficients as (1):

$$M_W = M_f + \alpha \times W \tag{1}$$

Here, M_W is the watermarked coefficients, M_f represents the filtered coefficients, and α is the insertion strength. α is adjusted to achieve the desired peak signal-to-noise ratio (PSNR) of 40 dB. Inverse DFT is applied to the watermarked image to obtain its luminance, and the final color image is then recovered using the unmodified chrominance.

2.2. Fourier-based detection

During the detection phase, only the watermarked image and the initial watermark are used, without the need for the original, unwatermarked image. Firstly, the luminance of the rectified image undergoes DFT processing. Next, coefficients are extracted from the magnitude along the predefined radius. Finally, the maximum normalized cross-correlation, denoted as C_{max}, is calculated between the original watermark W and the extracted DFT coefficients F as (2):

$$C_{max} = \max_{0\leq j\leq 1}\left(\frac{\sum_{i=0}^{N-1}(W(i)-\overline{W})(F(i+j)-\overline{F})}{\sqrt{\sum_{i=0}^{N-1}(W(i)-\overline{W})^2\sum_{i=0}^{N-1}(F(i+j)-\overline{F})^2}}\right) \tag{2}$$

Where N is the watermark length, $\overline{W}$ and $\overline{F}$ are respectively the means of the watermark sequence and the means of the extracted coefficients. If C_{max} exceeds a certain threshold t, the watermark is considered present.

2.3. Print-cam perspective correction

To improve the resilience of the watermarking system against perspective attacks during the print cam process, a corrective technique is carried out as a complementary measure. In this section, we present ReRNet, a neural network-based method to detect the corners of the wireframe around the ID image. Our approach addresses the challenge as a key point detection issue. These pivotal points correspond to the four corners of the image frame: the top left corner (TL), top right corner (TR), bottom right corner (BR), and bottom left corner (BL). An overview of the proposed correction method can be seen in Figure 2.

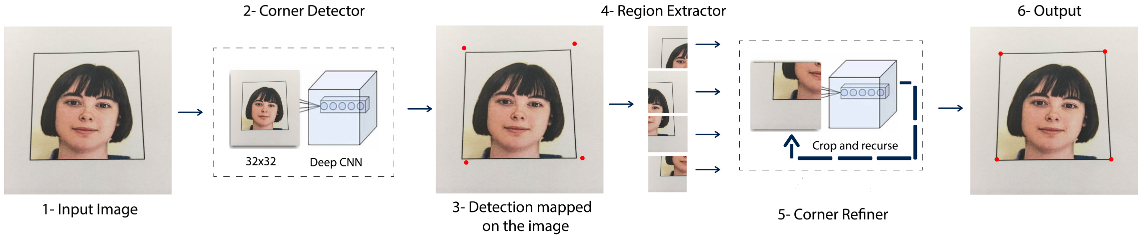


Figure 2. ReRNet architecture 32×32 image serves as input to the corner detector model, predicting four corners, which are then used by region extractor algorithm to obtain four corner images. Each of these corner images undergoes iterative refinement by the corner refiner model to produce the final output

In light of Figure 2, ReRNet can be divided into two main steps: initially, a deep convolutional neural network, modeled on the structure of ResNet-20, is employed to predict the four corners of the image collectively. Subsequently, each of these predictions is individually refined through the iterative application of the second convolutional neural network to predict one-corner coordinates. The architecture of ResNet-20 is depicted in Figure 3.

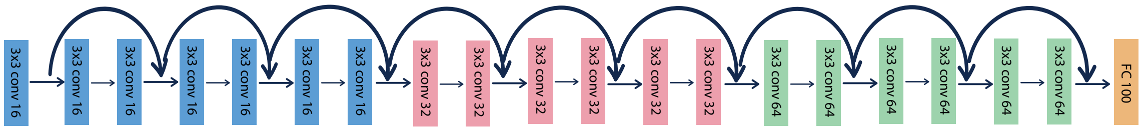


Figure 3. ResNet-20 architecture

2.3.1. Corner detector

In the first stage, the corner detector model is designed to predict the coordinates of the four corners of an input ID image, using the ResNet-20 network. As shown in Figure 3, ResNet-20 comprises sets of three convolutional blocks, each incorporating basic residual blocks. These blocks consist of two convolutional layers followed by batch normalization and rectified linear unit (ReLU) activation. Flattening the feature maps leads to a fully connected layer for regression tasks, with the final dense layer having a size of the output (default: 8 coordinates). Parameter initialization adheres to best practices, including He initialization for convolutional layers. The output of the model includes the predicted coordinates of the four corners, yielding a total of eight values. The structure of the corner detector model can be seen in Figure 4.

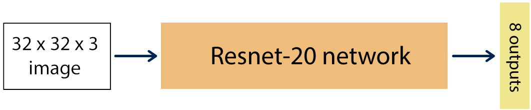


Figure 4. Corner detector structure

Ideally, we would like our model to achieve precise corner localization. However, this cannot be fully realized due to two fundamental challenges: features variation: the final convolutional layer is primarily designed to capture high-level features, which are effective for classification or bounding-box detection. However, these features might not generalize well for regression and precise localization tasks. Upscale error: upscaling input and output images can introduce a considerable ±10 pixel error, undermining the model's ability to precisely pinpoint the corners of interest.

Given this inherent imprecision, the output from the corner detector does not serve as the final result. Instead, it plays a crucial intermediate step in our approach. We leverage this output to extract four distinct regions from the image, ensuring that each region contains one of the four corners. To accomplish this, we have used a dedicated algorithm, aptly named region extractor, which effectively identifies and extracts these regions based on the corner detector's predictions.

Region extractor algorithm is used to identify and extract specific regions from the image based on predicted corner points labeled as TL', TR', BR' and BL'. The process involves cropping the image above and below each corner prediction along the x-axis and to the right and left along the y-axis. To focus on the region containing TL, for example, the method draws a line parallel to the y-axis through the middle of the x-coordinates of TL' and TR', thus eliminating the right-hand part of the image. Similarly, it excludes the area below a line parallel to the x-axis, intersecting the midpoint between of the y-coordinates of TL' and BL'. Where the area above and to the left of TL' exceeds that below or to the right, adjustments are made to balance them by removing a strip of the image on the top or left-hand side. It is important to note that this approach applies to the predictions of all four corners and ensures that the extracted regions are normalized to the image size.

2.3.2. Corner refiner

The four regions extracted using region extractor algorithm are then fed into the second model, referred to as the corner refiner, specially designed to locate a lone corner within images where only one corner of the frame is present. This model is also built upon the ResNet-20 architecture and features two regression heads, corresponding to one corner coordinates. Notably, it adopts the same parameters as the first model. Figure 5 gives an overview of the corner refiner structure.

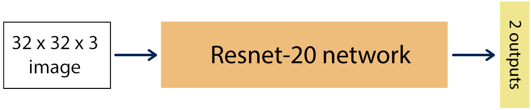


Figure 5. Corner refiner structure

However, due to the complexity of precisely locating the corner in a single step, the corner refiner model is used iteratively to converge on the accurate position of the corner. At each iteration, a portion of the image closest to a predicted point is retained in both vertical and horizontal directions based on a hyperparameter called β (retain factor), where $0 < \beta < 1$. As a result, after n iterations, the original (H, W) image is cropped down to $(H \times \beta^n, W \times \beta^n)$, and the process continues until the image size is smaller than (10×10), serving as the termination condition. Thanks to this recursive mechanism, we achieve precise corner detection with remarkable accuracy. Figure 6 shows the visual representation of this recursive corner refinement process.

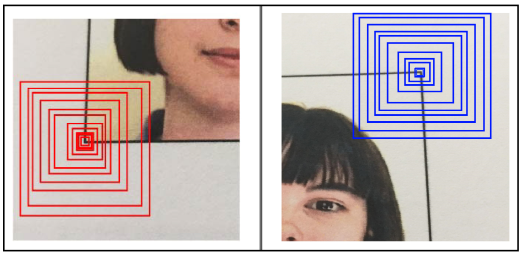


Figure 6. Corner retention process, based on the prediction, the system uses a predictive point to eliminate the zone least likely to contain the corner, then returns the cropped image to the model, with visually represented boxes indicating the areas observed at each iteration

2.3.3. Training setup

To prepare the training data, we started with 500 ID images sourced from PICS dataset [23]. These images were then printed and captured manually using iPhone 6 and Galaxy S5 smartphones with factory settings. The labeling of ground truth was done manually through the ImageJ application [24], using labels top left (TL), top right (TR), bottom right (BR), and bottom left (BL) to indicate specific corner positions. Subsequently, this dataset was augmented to a total of 16,000 images by applying random rotations, cropping, contrast adjustments, and changes in brightness [25]. For the first model, the captured images must contain four corners, whereas in the second model, these images are divided into four individual regions, with one corner included in each.

The input image is resized to 32×32 before being transmitted via the network. This helps to balance the trade-offs between computational efficiency, memory usage, and network performance. The stochastic gradient descent (SGD) is used with a learning rate of 0.05, momentum of 0.9, and weight decay of 0.00001. The training process spans 40 epochs, with the learning rate dynamically adjusted at epochs 10, 20, and 30. This process iterates through mini-batches of 32 samples, tracking the mean square error (MSE) loss. For validation, root mean squared error (RMSE) is used with evaluations carried out after each epoch.

2.3.4. Geometric correction

The process of capturing images freehandedly with a smartphone establishes geometric relationships between the camera position and the image [26]. This process of mapping the 3D world coordinate points onto the 2D plane of the image is known as projective transformation, often represented using matrix relations:

$$p = H \times P \tag{3}$$

Where p represents the 2D image coordinates, H stands for the 3×4 projective matrix describing the transformation, and P denotes the 3D world coordinates. In our scenario, where we capture an image from a 2D plane, the perspective transformation simplifies to:

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = H \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} \tag{4}$$

Where H is the 3×3 perspective matrix [9].

The H matrix has 8 degrees of freedom, requiring the estimation of 8 unknown variables. To determine the transformation that corrects the distorted image, the coordinates of the four corners are used to solve the system as (5):

$$\begin{cases} x = \frac{h_{11}X+h_{12}Y+h_{13}}{h_{31}X+h_{32}Y+1} \\ y = \frac{h_{21}X+h_{22}Y+h_{23}}{h_{31}X+h_{32}Y+1} \end{cases} \tag{5}$$

In summary, ReRNet is used to detect the four corners of the image frame. Then, the projective matrix is estimated by solving the system of equations mentioned in (5) using the corresponding four points. Ultimately, an inverse transformation is applied to the image to remap it into a rectified state.

3. RESULT AND DISCUSSION

In this section, we compare the enhanced print-cam watermarking method with its predecessor [22] in terms of detection rate. Both methods use Fourier watermarking to embed the watermark and incorporate perspective correction techniques to correct distortions introduced by the print-cam process. Gourrame *et al*. [22], hough transform was used to rectify the captured images, whereas our approach employs ReRNet for the same purpose.

The evaluation of the two methods was conducted under real print-cam conditions using 600 ID images from PICS dataset [23], 300 of them watermarked and the rest unwatermarked. These images were printed on paper using Konica Minolta Bizhub 450i, with dimensions of 44×44 mm. To capture these printed images, two mobile devices were used, a Redmi Note 10 and a Galaxy S21, with 64 and 108-megapixel cameras respectively. The acquisition process involved capturing the images freehandedly, without using any filters or flash during the capture. All images were taken under daylight illumination conditions. The test process is shown in Figure 7.

After perspective correction, the probability of true positive detection is calculated at different detection thresholds, and the results are shown in Figure 8. As seen from the figure, the proposed method exhibits remarkable results, achieving 80% in detection rate at a relatively high threshold of 0.3, surpassing the alternative method (42%) at the same threshold. This underscores the significant contribution of ReRNet in addressing perspective distortions during the print-cam process, ultimately improving the accuracy of watermark detection.

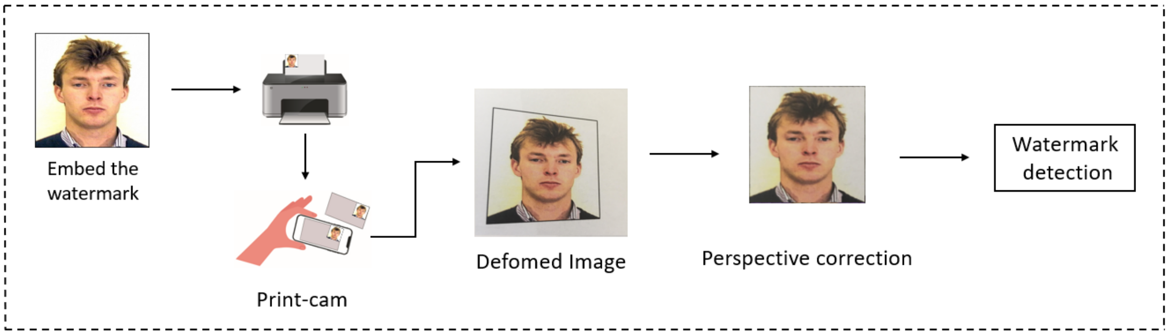


Figure 7. Real-world print-cam test procedure

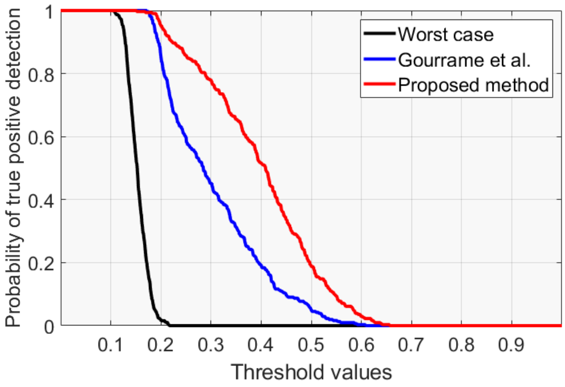


Figure 8. Probability of true positive detection at different threshold values under attacks (black line), and after perspective corrections using the proposed method (red line) and the method in [22] (blue line)

According to Figure 9, the proposed method demonstrates a remarkable enhancement in terms of detection rates, reaching 82% even at lower levels of probability of false alarm (PFA). Consequently, after ReRNet rectification, the probability of achieving true positive detection experiences a substantial increase, marking a notable improvement over its predecessor [22]. As a result, our proposed ReRNet-based watermarking technique shows outstanding performance, achieving a minimum error rate of 1.02%. In comparison, the other method exhibits a higher error rate of 1.37%. This confirms the advantage of our approach, which significantly reduces the error rate compared to the other method.

In summary, the insights from our study demonstrate the considerable influence of the suggested correction technique, ReRNet, in elevating the effectiveness of the Fourier watermarking approach during print-cam scenarios. The results show that ReRNet maintains high detection quality even under challenging attacks, meeting the industry benchmark error rate of 1%. This highlights its robustness and potential for implementation on smartphones.

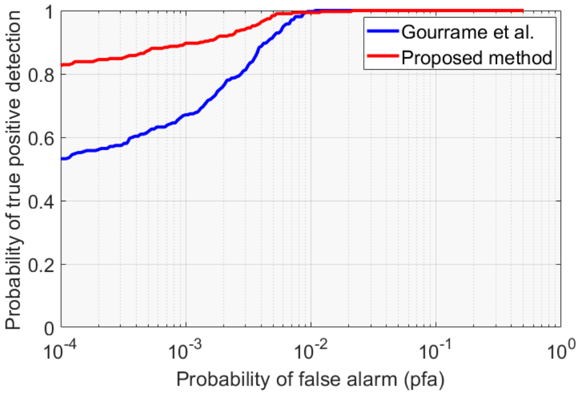


Figure 9. Detection rate in terms of receiver-operating characteristic (ROC) curves of the proposed approach and the method in [22]

4. CONCLUSION

In this paper, we present an improved Fourier-based image watermarking method for the print-cam process. The approach includes ReRNet, a new correction neural network to address perspective distortions often encountered in images freehandedly taken with a smartphone camera. Real-world testing was conducted to validate the effectiveness and reliability of our correction technique in an industrial setting applied to ID images. Results underscore the remarkable performance of the proposed scheme to successfully mitigate perspective distortions, a critical factor in a variety of print-cam applications. Additionally, our approach demonstrates improved detection rate values and shows a reduced error rate of 1.02% compared to 1.37% observed with its predecessor. The collective results from these assessments suggest that the improved Fourier watermarking method holds promise in addressing the challenges presented during the print-cam process and contributes to the advancement of the digital image watermarking field.

REFERENCES

[1] S. Pouyanfar, Y. Yang, S.-C. Chen, M.-L. Shyu, and S. S. Iyengar, "Multimedia big data analytics," *ACM Computing Surveys*, vol. 51, no. 1, pp. 1–34, Jan. 2019, doi: 10.1145/3150226.

[2] S. P. Mohanty, A. Sengupta, P. Guturu, and E. Kougianos, "Everything you want to know about watermarking: from paper marks to hardware protection," *IEEE Consumer Electronics Magazine*, vol. 6, no. 3, pp. 83–91, Jul. 2017, doi: 10.1109/MCE.2017.2684980.

[3] K. Gourrame, F. Ros, H. Douzi, R. Harba, and R. Riad, "Fourier image watermarking: print-cam application," *Electronics*, vol. 11, no. 2, p. 266, Jan. 2022, doi: 10.3390/electronics11020266.

[4] V. Licks and R. Jordan, "Geometric attacks on image watermarking systems," *IEEE Multimedia*, vol. 12, no. 3, pp. 68–78, Jul. 2005, doi: 10.1109/MMUL.2005.46.

[5] A. Pramila, A. Keskinarkaus, and T. Seppänen, "Camera based watermark extraction-problems and examples," in *Proceedings of the Finnish Signal Processing Symposium*, 2007.

[6] L. Rakhmawati, W. Wirawan, and S. Suwadi, "A recent survey of self-embedding fragile watermarking scheme for image authentication with recovery capability," *EURASIP Journal on Image and Video Processing*, vol. 2019, no. 1, p. 61, Dec. 2019, doi: 10.1186/s13640-019-0462-3.

[7] M. Begum and M. S. Uddin, "Digital image watermarking techniques: a review," *Information*, vol. 11, no. 2, p. 110, Feb. 2020, doi: 10.3390/info11020110.

[8] K. Rangel-Espinoza, E. Fragoso-Navarro, C. Cruz-Ramos, R. Reyes-Reyes, M. Nakano-Miyatake, and H. M. Pérez-Meana, "Adaptive removable visible watermarking technique using dual watermarking for digital color images," *Multimedia Tools and Applications*, vol. 77, no. 11, pp. 13047–13074, Jun. 2018, doi: 10.1007/s11042-017-4931-3. 13 047–13 074, 2018.

[9] T. Huynh-The *et al*., "Selective bit embedding scheme for robust blind color image watermarking," *Information Sciences*, vol. 426, pp. 1–18, Feb. 2018, doi: 10.1016/j.ins.2017.10.016.

[10] Sunesh and R. R. Kishore, "A novel and efficient blind image watermarking in transform domain," *Procedia Computer Science*, vol. 167, pp. 1505–1514, 2020, doi: 10.1016/j.procs.2020.03.361.

[11] T. Nakamura, A. Katayama, M. Yamamuro, and N. Sonehara, "Fast watermark detection scheme for camera-equipped cellular phone," in *Proceedings of the 3rd International Conference on Mobile and Ubiquitous Multimedia*, New York, NY, USA: ACM, Oct. 2004, pp. 101–108. doi: 10.1145/1052380.1052395.

[12] L. Dong, J. Chen, C. Peng, Y. Li, and W. Sun, "Watermark-preserving keypoint enhancement for screen-shooting resilient watermarking," in *2022 IEEE International Conference on Multimedia and Expo (ICME)*, IEEE, Jul. 2022, pp. 1–6. doi: 10.1109/ICME52920.2022.9859950.

[13] A. Pramila, A. Keskinarkaus, V. Takala, and T. Seppänen, "Extracting watermarks from printouts captured with wide angles using computational photography," *Multimedia Tools and Applications*, vol. 76, no. 15, pp. 16063–16084, Aug. 2017, doi: 10.1007/s11042-016-3895-z.

[14] Z. Wang et al., "Data hiding with deep learning: a survey unifying digital watermarking and steganography," *IEEE Transactions on Computational Social Systems*, vol. 10, no. 6, pp. 2985–2999, 2023, doi: 10.1109/TCSS.2023.3268950.

[15] M. Tancik, B. Mildenhall, and R. Ng, "StegaStamp: invisible hyperlinks in physical photographs," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jun. 2020, pp. 2114–2123. doi: 10.1109/CVPR42600.2020.00219.

[16] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "BiSeNet: bilateral segmentation network for real-time semantic segmentation," in *Computer Vision – ECCV 2018. ECCV 2018. Lecture Notes in Computer Science()*, Cham: Springer, 2018, pp. 334–349. doi: 10.1007/978-3-030-01261-8_20.

[17] S. Boujerfaoui, H. Douzi, R. Harba, and F. Ros, "Cam-Unet: print-cam image correction for zero-bit fourier image watermarking," *Sensors*, vol. 24, no. 11, p. 3400, May 2024, doi: 10.3390/s24113400.

[18] S. Boujerfaoui, H. Douzi, and R. Harba, "Print-cam image distortion correction for robust image watermarking," in *2024 26th International Conference on Digital Signal Processing and its Applications (DSPA)*, IEEE, Mar. 2024, pp. 1–6. doi: 10.1109/DSPA60853.2024.10510115.

[19] X. Zhong, P.-C. Huang, S. Mastorakis, and F. Y. Shih, "An automated and robust image watermarking scheme based on deep neural networks," *IEEE Transactions on Multimedia*, vol. 23, pp. 1951–1961, 2021, doi: 10.1109/TMM.2020.3006415.

[20] J. Jia *et al*., "RIHOOP: robust invisible hyperlinks in offline and online photographs," *IEEE Transactions on Cybernetics*, vol. 52, no. 7, pp. 7094–7106, Jul. 2022, doi: 10.1109/TCYB.2020.3037208.

[21] R. Riad, F. Ros, R. Harba, H. Douzi, and M. El Hajji, "Pre-processing the cover image before embedding improves the watermark detection rate," in *2014 Second World Conference on Complex Systems (WCCS)*, IEEE, Nov. 2014, pp. 705–709. doi: 10.1109/ICoCS.2014.7060967.

[22] K. Gourrame *et al*., "A zero-bit fourier image watermarking for print-cam process," *Multimedia Tools and Applications*, vol. 78, no. 2, pp. 2621–2638, Jan. 2019, doi: 10.1007/s11042-018-6302-0.

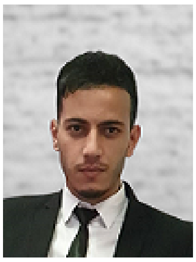
[23] P. Hancock, "Psychological image collection at stirling (PICS)." https://pics.stir.ac.uk/, [Online] (accessed on Jan 23, 2022).

[24] C. A. Schneider, W. S. Rasband, and K. W. Eliceiri, "NIH image to ImageJ: 25 years of image analysis," *Nature Methods*, vol. 9, no. 7, pp. 671–675, Jul. 2012, doi: 10.1038/nmeth.2089.

[25] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, no. 1, Dec. 2019, doi: 10.1186/s40537-019-0197-0.

[26] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge University Press, 2004. doi: 10.1017/CBO9780511811685

BIOGRAPHIES OF AUTHORS



Said Boujerfaoui Ph.D. student at Ibn Zohr University in Morocco. He obtained his master's degree in data science from the same university in 2020. His areas of research include artificial intelligence, image processing, digital watermarking and computer vision. He can be contacted at email: said.boujerfaoui@edu.uiz.ac.ma.

Hassan Douzi received the Doctorat (French Ph.D.) from The University of Paris IX (Dauphine) in application of wavelets to seismic inversion problem. Since 1993 he is a Research Professor at the University Ibn Zohr of Agadir in Morocco. His research interests include wavelets, image and signal processing. He can be contacted at email: h.douzi@uiz.ac.ma.

Rachid Harba graduated from ENS Cachan in electrical engineering, Paris, France, in 1983, and obtained his Ph.D. in electrical engineering in 1985 from INPG Grenoble, France. In 1988, he became an associate professor at the University of Orléans, France, and is currently a professor at the University of Orléans, in the Polytech'Orléans department. His research focuses on the statistical processing of signals and images, particularly for medical and industrial applications. Currently, he is the leader of the European project STANDUP #777661 for the early detection of diabetic foot using a smartphone. He can be contacted at email: rachid.harba@univ-orleans.fr.

Khadija Gourrame holds a Ph.D. in computer science from University of Ibn Zohr in Morocco and University of Orleans in France, under a cotutelle scheme. She received a master's degree in computer science from university of Mysore, India, in 2014. Her research interests include image processing, digital watermarking, and computer vision. She can be contacted at email: khadija.gourrame@edu.uiz.ac.ma.