An Automated and Robust Image Watermarking Scheme Based on Deep Neural Networks

Xin Zhong, Pei-Chi Huang, Spyridon Mastorakis, Frank Y. Shih, *Senior Member, IEEE*

***Abstract*—Digital image watermarking is the process of embedding and extracting a watermark covertly on a cover-image. To dynamically adapt image watermarking algorithms, deep learning–based image watermarking schemes have attracted increased attention during recent years. However, existing deep learning–based watermarking methods neither fully apply the fitting ability to learn and automate the embedding and extracting algorithms, nor achieve the properties of robustness and blindness simultaneously. In this paper, a robust and blind image watermarking scheme based on deep learning neural networks is proposed. To minimize the requirement of domain knowledge, the fitting ability of deep neural networks is exploited to learn and generalize an automated image watermarking algorithm. A deep learning architecture is specially designed for image watermarking tasks, which will be trained in an unsupervised manner to avoid human intervention and annotation. To facilitate flexible applications, the robustness of the proposed scheme is achieved without requiring any prior knowledge or adversarial examples of possible attacks. A challenging case of watermark extraction from phone camera–captured images demonstrates the robustness and practicality of the proposal. The experiments, evaluation, and application cases confirm the superiority of the proposed scheme.**

***Index Terms*—Image watermarking, automation, robustness, deep learning, convolutional neural networks.**

I. Introduction

Digital image watermarking refers to the process of embedding and extracting information covertly on a cover-image. The data (*i.e.,* the watermark) is hidden into a cover-image to create a to-be-transmitted marked-image. The marked-image does not visually reveal the watermark, and only the authorized recipients can extract the watermark information correctly. The techniques of image watermarking can be applied for various applications. Based on different target scenarios, the watermark information can be presented in different forms; for example, the watermark can be some random bits or electronic signatures for image protection and authentication [1], or some hidden messages for covert communication [2]. In addition, the watermark can be encoded for different purposes, such as added security with encryption methods or restoring information integrity with error correction codes during a cyberattack [3].

For copyright protection, classic image watermarking research [4] only focuses on single-bit extractions, where the output indicates whether an image contains a watermark or not. To enable a wide range of applications, modern image watermarking research primarily focuses on multi-bit scenarios that extract the entire communicative watermark information [3], [5]. Typically, many factors should be considered in an image watermarking scheme, such as the fidelity of the marked-image and the watermark's undetectability to computer analysis. The proposed image watermarking scheme not only satisfies these factors but achieves the robustness of its priority: the watermark should survive even if the marked-image is degraded or distorted. Ideally, a robust image watermarking scheme keeps the watermark intact under a designated class of distortion without any assistance of techniques. However, in practice, the watermark is extracted approximately in many attacking scenarios, and various encoding methods can be applied for its restoration [6]. Achieving robustness is a major challenge in a blind image watermarking scheme, where the extraction must be performed without any information about the original cover-image.

Due to the limited scope of manual design, the traditional image watermarking methods encounter difficulties; for example, extraction can only tolerate certain types of distortions in robust watermarking schemes, or the watermark itself can only resist a limited range of computer analysis in undetectable watermarking schemes [7]. To break through these drawbacks, incorporating deep learning into image watermarking has attracted increased attention in recent years [8]. Deep learning, as a representation learning method, has enabled significant improvements in computer vision through its ability to fit and generalize complex features. The major advantage of deep learning methodologies for image watermarking is that they perform image watermarking in a more adaptive manner by dynamically learning algorithms to extract both high- and low-level features for image watermarking from multiple instance big data.

Recent research on image watermarking tasks with deep neural networks has emerged [9], [10], [11], [12], but there still exist challenging issues. For example, it is difficult to fully utilize the fitting ability of deep neural networks to automatically learn and generalize both the watermark embedding and extracting processes. Also, labeling the ground truth for an image watermarking task can be ill-defined or time-consuming. Finally, achieving robustness and blindness simultaneously without prior knowledge of adversarial examples [12] remains unexplored.

To address the above challenges, we present an automated, blind, and robust image watermarking scheme based on deep learning neural networks. The contribution of this paper is threefold. First, the fitting ability of deep neural networks is ex-

Xin Zhong, Pei-Chi Huang and Spyridon Mastorakis are with Department of Computer Science, University of Nebraska Omaha, Omaha, NE, 68182 USA e-mail: {xzhong, phuang, smastorakis}@unomaha.edu

Frank Y. Shih is with Department of Computer Science, New Jersey Institute of Technology, Newark, NJ 07102 USA e-mail: shih@njit.edu

This paper has been accepted for publication by the IEEE Transactions on Multimedia. The copyright is with the IEEE. DOI: 10.1109/TMM.2020.3006415

ploited to automatically learn image watermarking algorithms, facilitating an automated system without requiring domain knowledge. Second, the proposed deep learning architecture can be trained in an unsupervised manner to reduce human intervention, which is suitable for image watermarking. Finally, experimental results demonstrate the robustness and accuracy of the proposed scheme without using any prior knowledge or adversarial examples of possible attacks.

The remainder of this paper is organized as follows. The related work is described in Section II. The proposed scheme is presented in Section III. Experiments and analysis are presented in Section IV. Applications of the proposed watermarking scheme are discussed in Section V. Finally, conclusions are drawn in Section VI.

II. Related Work

This section provides a detailed analysis of recent reports. Table I shows the analytical comparison of our proposed scheme with the start-of-the-art deep learning–based image watermarking schemes.

In handcrafted watermarking algorithms, various optimization methods have been applied to adapt the embedding parameters, and this research direction has attracted attentions in recent years [13], [14], [15]. Consequently, exploring the optimization ability of deep learning models for adaptive and automated image watermarking is of great interest. However, compared to significant advancements on image steganography with deep neural networks [16], [17], deep learning–based image watermarking is still in its infancy.

Kandi *et al.* [8] applied two convolutional autoencoders to reconstruct a cover-image. In a marked-image, the pixels produced by the first autoencoder indicate bits with the value of zero, and the pixels produced by the second autoencoder indicate bits with the value of one, hence developing a non-blind binary watermarking scheme. Vukotić *et al.* [9] developed a deep learning–based, single-bit watermarking scheme by embedding through designed adversarial images and extracting via the first layer of a trained deep learning model. Li *et al.* [10] embedded a watermark into the discrete cosine domain by traditional algorithms [4] and applied convolutional neural networks to facilitate the extraction.

Besides the single-bit and multi-bit watermarking, attempts were also reported in special scenarios. For example, for zero-watermarking, where a master share is sent separately from the image, Fierro-Radilla *et al.* [11] applied convolutional neural networks to extract the required feature from the cover-image and linked these features with the watermark to create a master share. For the scenario of template-based watermarking, Kim *et al.* [18] embedded a handcrafted template by using a classic additive method and estimated possible distortion parameters by comparing the extracted template to the original one with the help of convolutional neural networks. Thus far, the existing deep learning–based image watermarking schemes do not fully apply the fitting ability of deep neural networks to learn and generalize the embedding and extracting algorithms.

Furthermore, due to the fragility of deep neural networks [12], a modified image as an input to a trained deep learning model may cause failure. In other words, robustness is a major challenge in deep–learning based image watermarking because noise or modifications of the marked-image can result in extraction failures. Mun *et al.* [19] proposed to solve this issue by proactively including noisy marked-images as adversarial examples in the training phase. However, enumerating all types of attacks and their combinations may not be practically feasible.

To the best of our knowledge, our proposed scheme is the first method that explores the ability of deep neural networks in automatically learning and generalizing both watermark embedding and extracting algorithms, while achieving robustness and blindness simultaneously.

III. Proposed Image Watermarking Scheme

We revisit the typical design of an image watermarking scheme and present the overview architecture of our scheme in Section III-A. Then, we present the loss function design and the scheme objective in Section III-B. Finally, the detailed structure of the proposed model is described in Section III-C.

A. The Overview Architecture of the Proposed Scheme

The traditional design of an image watermarking scheme is shown in Fig.1. A watermark (denoted as w) is embedded into a cover-image (denoted as c) to produce a marked-image (denoted as m) that looks similar to c and is transported through a communication channel. Then, the receiver extracts the watermark data (denoted as w^*) from the received marked-image (denoted as m^*) that could be a modified version of m if some distortions or attacks are occurred during the transmission.

To embed w into c, typically, the first step is to project c into one of its feature spaces in spatial, frequency, or other domains. Next, w is encoded and embedded into the feature space of c. The embedded feature space is projected back into the cover-image space to create a marked-image m. Inversely, the watermark extraction is to project the marked-image reception m^* to the same feature space and then extract and decode the watermark information. Based on different target applications, traditional image watermarking methods manually design the projection, embedding, extraction, encoding, and decoding functions. As the criteria of a design, an image watermarking scheme often highlights its fidelity (*i.e.,* high similarity between the m and c) and robustness (*i.e.,* keeping the integrity of w^* when m^* is distorted).

Traditional image watermarking methods perform competently through hand-designed algorithms; however, it remains challenging to automatically learn these algorithms without complete dependence upon careful design. To tackle this difficulty, we propose a novel scheme which develops a deep learning model to automatically learn and generalize the embedding, the extraction, and the invariance functions in image watermarking. Fig. 2 illustrates the overall architecture of the proposed scheme with some example images. Given two input spaces that are all the possible inputs of watermark images and cover-images (W and C, respectively), neural network μ_{θ_1} parameterized by θ_1 is applied to learn a function that encodes W. W_f, the encoded space of W, not only enlarges

TABLE I: An analytical comparison between the proposed scheme and state-of-the-art image watermarking methods applying deep neural networks.

Method	Learning Watermarking Algorithms?	Blind	Robust	Extraction Type
Kandi *et al.* [8]	No	No	Robust to common image processing attacks	Multi-bit
VukotiÄ *et al.* [9]	Learning extraction	Yes	Robust to Rotation, JPEG, and Cropping	Single-bit
Li *et al.* [10]	No	No	No	Multi-bit
Fierro-Radilla *et al.* [11]	No	No	Robust to common image processing attacks	Zero-watermarking
Kim *et al.* [18]	Assisting extraction	No	Focus on geometric attacks	Template-based watermarking
Mun *et al.* [19]	Learning extraction	Yes	Robust to all enumerated attacks during training	Multi-bit
Ours	Yes	Yes	Robust to common image processing attacks	Multi-bit

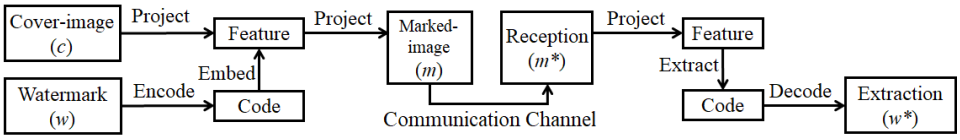

Fig. 1: The traditional design of an image watermarking scheme.

W to prepare for the next-step concatenation, but also brings some redundancy, decomposition, and perceivable randomness to help information protection and robustness. Like the embedding process in traditional watermarking where an encoded w is inserted into a feature space of c, in the proposed scheme, an embedder that takes W_f, C as inputs and produces the marked-image is fit by the neural network σ_{θ_2} parameterized by θ_2. The marked-image space is named as M. To handle possible distortions, a neural network τ_{θ_5} parameterized by θ_5 is introduced to learn to convert M to its enlarged and redundant transformed-space T. After the transformation, T preserves information about W_f and rejects other irrelevant information, such as noises on M, therefore providing robustness. Finally, the inverse watermark reconstruction functions are fitted by two neural network components, φ_{θ_3} and γ_{θ_4} with trainable parameters θ_3 and θ_4, that extract W_f from T and decode W from W_f, respectively.

To compare the proposed scheme in Fig. 2 with the traditional design in Fig.1, one can observe that the neural network components fit and optimize the image watermarking process dynamically. Encoder and decoder networks (denoted as μ_{θ_1} and γ_{θ_4}, respectively) fit the watermark encoding and decoding functions. An embedder (denoted as σ_{θ_2}) projects C into a feature space, embeds W_f into the space, and projects to the marked-image space M. An extractor (denoted as φ_{θ_3}) inverses all the processes in σ_{θ_2}, and τ_{θ_5} handles the distortion during the transmission through a communication channel. More details of the architectures are described in Section III-C.

Compared to an autoencoder [20], where an input space is transformed to a representative and intermediate feature space and the original input is recovered from this feature space, the proposed scheme takes two spaces C and W as inputs, produces an intermediate marked-image space M, and recovers W from M. The recovery ability of autoencoders, *i.e.,* an exact reconstruction of the input with appropriate features extracted by the deep neural networks, secures the feasibility of watermark extraction in the proposed scheme. The reconstruction requires only M without the need of W and C, enabling the *blindness* of the proposed scheme. A feature space in an autoencoder is often learned through a bottleneck for the dimensionality compression, but the proposed scheme learns equal-sized or over-complete representations to achieve high *robustness* and *accuracy* of watermark extraction.

B. The Loss Function and Scheme Objective

The entire architecture is trained as a single deep neural network with several loss terms designed for image watermarking. Given the data samples $w_i \in W$, $i = 1, 2, 3, \ldots$ and $c_i \in C$, $i = 1, 2, 3, \ldots$, the proposed scheme can be trained in an unsupervised manner. There are two inputs w_i and c_i, and two outputs w_i^* and m_i in the proposed deep neural network. For the output w_i^*, an extraction loss that minimizes the difference between w_i^* and w_i is computed to ensure full extraction of the watermark. For the output m_i, a fidelity loss that minimizes the difference between m_i and c_i is computed to enable watermarking invisibility. For the output m_i, we also compute an information loss that forces m_i to contain the information of w_i. To achieve this, we maximize the correlation between feature maps of w_f^i and feature maps of m_i, where the feature maps are the outputs of convolutional layers in the proposed architecture, and the feature maps of w_f^i and m_i, *i.e.*, B_1 and B_2, are illustrated in Figs. 3 and 4. Denoting the parameters to be learned as $\vartheta = [\theta_1, \theta_2, \theta_3, \theta_4, \theta_5]$, the loss function $L(\vartheta)$ of the proposed scheme can be expressed as:

$$L(\vartheta) = \lambda_1 \|w_i^* - w_i\|_1 + \lambda_2 \|m_i - c_i\|_1 + \lambda_3 \psi(m_i, w_f^i), \quad (1)$$

where λ_i, $i = 1, 2, 3$ is the weight factor and ψ is a function computing the correlation given as:

$$\psi(m_i, w_f^i) = \frac{1}{2}(\|g(B_1(w_f^i)), g(B_1(m_i))\|_1 + \|g(B_2(w_f^i)), g(B_2(m_i))\|_1), \quad (2)$$

where g denotes the Gram matrix that contains all possible inner products. By minimizing the distance between the Gram matrices of the feature maps of m_i and w_f^i produced by intermediate layer outputs B_1 and B_2, we maximize their correlation. As the feature producers, the annotation of B_1 and B_2 is presented in Fig. 4, and the convolution block B that

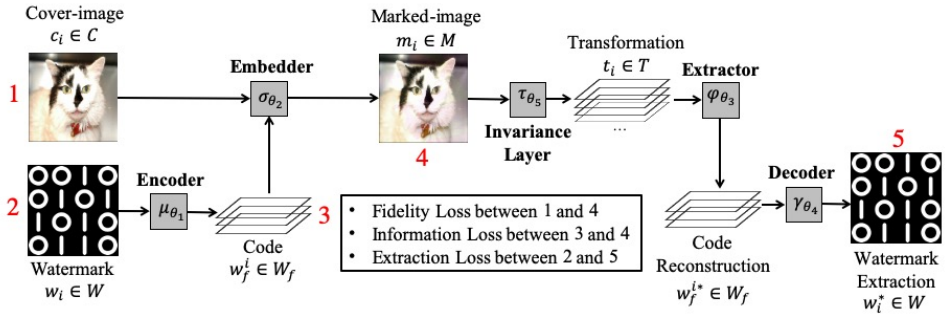

Fig. 2: The overall architecture of the proposed image watermarking scheme.

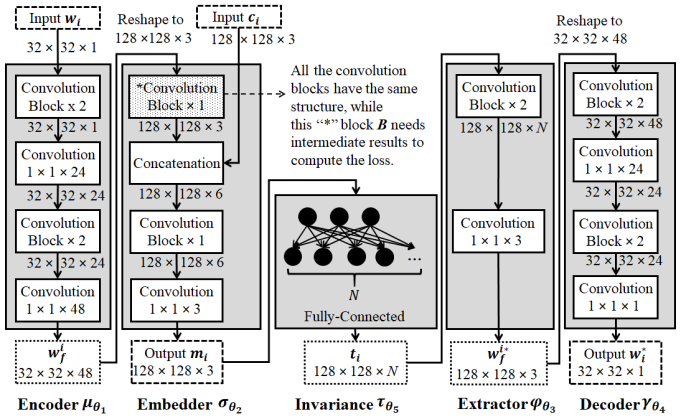

Fig. 3: The detailed components of the proposed watermarking scheme: the Encoder μ_{θ_1}, the Embedder σ_{θ_2}, the Invariance Layer τ_{θ_5}, the Extractor φ_{θ_3}, and the Decoder γ_{θ_4}. Every structure of the convolution block is the same, but only block marked with "*" needs intermediate results to compute the loss.

contains B_1 and B_2 is annotated in Fig. 3. More discussions will be presented in Section III-C.

In Eq. 1, each two of the fidelity loss, information loss, and extraction loss terms can be a trade-off for image watermarking. For example, minimizing the fidelity loss term to zero means that m_i is identical to c_i. However, there is no embedded information in m_i in this case, so the extraction of w_i will fail. To allow some imperfectness of the loss terms, the mean absolute error (*i.e.,* the L1 norm) is selected to highlight the overall performance rather than a few outliers.

With regularization, the proposed scheme objective is represented as $L(\vartheta)+\lambda_4 P$, where P is the penalty term to achieve robustness as in Eq. 6, and λ_4 is the weight controlling the strength of the regularization term. The deep neural network needs to learn the parameter ϑ^* that minimizes $L(\vartheta)+\lambda_4 P$:

$$\vartheta^* = argmin_\vartheta [L(\vartheta) + \lambda_4 P]. \tag{3}$$

In the backpropagation during training, the term $\lambda_1 \|w_i^* - w_i\|_1$ is applied by all the components of the proposed architecture in their weight updates, while only μ_{θ_1} and σ_{θ_2} apply terms $\lambda_2 \|m_i - c_i\|_1$ and $\lambda_3 \psi(m_i, w_f^i)$ to their weight updates. This enables μ_{θ_1} and σ_{θ_2} to encode and embed the information in a way that φ_{θ_3} and γ_{θ_4} are able to extract and decode the watermark.

C. Detailed Structure of the Proposed Neural Networks Model

This subsection describes the major components design of neural networks: μ_{θ_1}, σ_{θ_2}, φ_{θ_3}, γ_{θ_4} and τ_{θ_5} in more detail. The overall design is modularized and illustrated in Fig. 3. If we single out two pairs (μ_{θ_1}, γ_{θ_4}) and (σ_{θ_2}, φ_{θ_3}), we can find that each pair is conceptually symmetrical. The watermark is considered as a $32\times32\times1$ binary image, and the cover-image is a $128\times128\times3$ color image in this description. One might adapt and customize the image sizes based on different target applications.

1) The Encoder μ_{θ_1} and the Decoder γ_{θ_4}: Given the samples w_i, $i = 1, 2, 3, ...$ from the input space W, the encoder μ_{θ_1} learns a function that encodes W to its code W_f. Inversely, the decoder γ_{θ_4} learns the decoding function from W_f to W with samples w_f^{i*}, $i = 1, 2, 3, ...$ The encoder μ_{θ_1} successively increases a $32\times32\times1$ binary watermark image to $32\times32\times24$ and $32\times32\times48$, and the decoder γ_{θ_4} successively decreases the $32\times32\times48$ feature space back to a $32\times32\times1$ binary watermark image. The reason to train this channel-wise increment is two-fold. First, it produces a $128\times128\times3$ w_f^i that has the same width and height as the cover-image, so that we can concatenate a feature map of w_f^i and c_i along their channel dimension. Each of w_f^i and c_i will contribute equally to the $128\times128\times6$ concatenated matrix used in the embedder σ_{θ_2}. Thus, we are evenly weighing the watermark and the cover-image. Second, this $32\times32\times1$ to $32\times32\times48$ increment introduces some redundancy, decomposition, and perceivable randomness to W, which helps information protection and robustness.

2) The Embedder σ_{θ_2} and the Extractor φ_{θ_3}: The embedder σ_{θ_2} applies the convolution block B to extract a $128\times128\times3$ to-be-embedded feature map of w_f^i that is concatenated along the channel dimension with the cover-image. Directly applying c_i, while only applying a feature map of w_f^i, helps c_i to dominate the appearance. The $128\times128\times6$ concatenation is fed into another convolution block to produce m_i. The extractor φ_{θ_3} inverses the process by two successive convolution blocks.

To capture various scales of features for image watermarking, the inception residual block [21] is applied. It consists of a 1×1, a 3×3, and a 5×5 convolution, as well as a residual connection that sums up the features and the input itself. In the proposed structure, each convolution has 32 filters, and the 5×5 convolution is replaced by two 3×3 convolutions for efficiency. The 32-channel feature maps produced by different convolution paths are concatenated along

the channel dimension to form a 96-channel feature, and a 1 × 1 convolution is applied to convert the 96-channel feature back to the original input channel size for the summation in the residual connection. The architecture of a convolution block is shown in Fig. 4, where F_w, F_d, and F_c, respectively, denote the size of height, width, and channel.

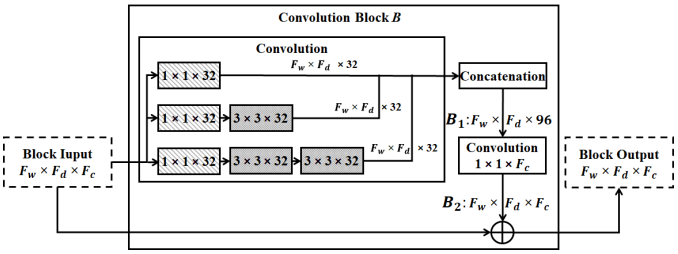

Fig. 4: The architecture design of a convolution block. The block input/output size is denoted as the size of height (F_w), the width (F_d), and the channel (F_c), respectively.

All the convolution blocks in Fig. 3 have the same inception residual structure as shown in Fig. 4. In the case of the "*" convolution block B of Fig. 3, the annotated intermediate results B_1 and B_2 of Fig. 4 are applied in Eq. 2. Specifically, block B extracts features not only from its input w_f^i in the architecture, but also from m_i. The annotated $F_w \times F_d \times 96$ and $F_w \times F_d \times F_c$ feature maps are the intermediate results B_1 and B_2, respectively.

3) The Invariance Layer τ_{θ_5}: The invariance layer is the key component to provide robustness in the proposed image watermarking scheme. Using a fully-connected layer, τ_{θ_5} learns a transformation from space M to an over-complete space T, where the neurons are activated sparsely. The idea is to redundantly project the most important information from M into T and to deactivate the neural connections of the areas on M irrelevant of the watermark, thus preserving the watermark even if there is noise or distortion that modified a part of M. As shown in Fig. 3, τ_{θ_5} converts a 3-color-channel instance m_i of M into an N-channel ($N \geq 3$ for non-compression) instance t_i of T, where N is the redundant parameter. Increasing N results in increased redundancy and decomposition in T, which provides higher tolerance of the errors in M and enhances robustness.

Based on the contractive autoencoder [22], τ_{θ_5} employs a regularization term that is obtained by the Frobenius norm of the Jacobian matrix of a layer's outputs with regards to its inputs. Mathematically, the regularization term P is given as:

$$P = \Sigma_{i,j} \left(\frac{\partial h_j(X)}{\partial X_i} \right)^2, \tag{4}$$

where X_i denotes the i-th input and h_j denotes the output of the j-th hidden unit of the fully connected layer. Similar to a common gradient computation, the Jacobian matrix can be written as:

$$\frac{\partial h_j(X)}{\partial X_i} = \frac{\partial A(\omega_{ji} X_i)}{\partial \omega_{ji} X_i} \omega_{ji}, \tag{5}$$

where A is an activation function and ω_{ji} is the weight between h_j and X_i. We set A as the hyperbolic tangent (tanh) for strong gradients and bias avoidance [23], and hence P can be computed as:

$$P = \Sigma_j (1 - h_j^2)^2 \Sigma_i (\omega_{ji}^T)^2. \tag{6}$$

If the value of P is minimized to zero, all weights ω in τ_{θ_5} will be zero, so that the output of τ_{θ_5} will be always zero no matter how we change the inputs X. Thus, minimizing P alone will cause the rejection to all the information from the inputs m_i. Therefore, we place P as a regularization term in the total loss function to preserve useful information related to the loss terms of image watermarking, while rejecting all other noise and irrelevant information. In this way, we achieve robustness without prior knowledge of possible distortion.

Remarkably, each color channel in m_i is treated as a single input unit to significantly improve the computational efficiency. For example, if we treat one pixel as an input, a $128 \times 128 \times 3$ marked-image will have 49,152 input units. Setting the redundant parameter N to its smallest value 3 will imply $49,152 \times 3$ (= 147,456) units in the fully-connected invariance layer τ_{θ_5}, which requires at least $49,152 \times 147,456$ (= 7,247,757,312) parameters. This significantly lowers the efficiency. On the other hand, treating one color channel as an input unit considers only 3 input units for an RGB marked-image, which enables faster computation with much fewer parameters as well as a much larger N to enable higher redundancy for higher robustness.

IV. Experiments and Analysis

This section experimentally analyzes quantitative and analytical evaluation of the proposed deep learning–based image watermarking scheme. Section IV-A introduces our data preparation, and Section IV-B presents the experimental design, training and validation. To validate our proposed image watermarking approach, Section IV-C provides special testing experiments on synthetic images, and Section IV-D shows the robustness in different distortion. A feasibility case study on the scenario of watermark extraction from phone camera pictures is also presented in Section IV-E.

A. Preparation of Datasets

The proposed deep learning–based image watermarking architecture was trained as a single deep neural network. ImageNet [24] was rescaled to size 128×128 with RGB channels and then used as the cover-images. The binary version of CIFAR [25] with its original size 32×32 was used as the watermarks because the proposed architecture used 32×32 binary watermark images. Both datasets contain more than millions of images such that the proposed scheme during training can be introduced by a large scope of instances. For a validation set after each training epoch, 10,000 images from each dataset that were not used during the training phase are separated.

The testing is performed on 10,000 images (rescaled to 128×128) from the Microsoft COCO dataset [26] as the cover-images, and 10,000 images of the testing set of the binary CIFAR as the watermarks. Both the testing cover-images and watermarks were not applied in the training to demonstrate that

the proposed scheme learns and generalizes the watermarking algorithms without over-fitting to the training samples.

B. Training, Validation and Testing of the Proposed Model

As described in Section III, the proposed image watermarking scheme is trained as a single and deep neural network. The ADAM optimizer [27], which adopts a moving window in the gradient computation, is applied, for its ability of continuous learning after large epochs. The training and validation of the proposed scheme are shown in Fig. 5, where the values of the terms in the loss (Eq. 1) and objective (Eq. 3) during 200 epochs are presented. During both training and validation, the terms T1 and T2 (defined in Fig. 5) in $L(\vartheta)$ converge smoothly below 0.015, and $L(\vartheta)+\lambda_4 P$ converges below 0.03, indicating a proper fit. Term T1 has slightly more errors because when carrying the watermark, a marked-image cannot be completely identical to a cover-image. λ_1, λ_2, and λ_3 are all set to be 1 to equally weigh the terms, and λ_4 is set to be 0.01 as suggested in [22]. All the layers apply the rectified linear unit (ReLU) as the activation function apart from the outputs (marked-image and watermark extraction), which use sigmoid to limit the range to (0, 1).

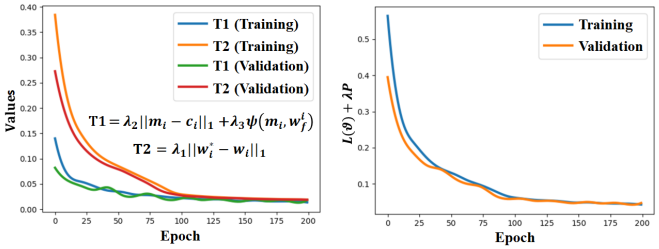

Fig. 5: The loss values during training and validation.

At the testing phase, the peak signal-to-noise ratio (*PSNR*) and bit-error-rate (*BER*) are respectively used to quantitatively evaluate the fidelity of the marked-images and the quality of the watermark extraction. The *PSNR* is defined as:

$$PSNR = 10\log_{10}\left(\frac{max(c_i)^2}{MSE(c_i, m_i)}\right), \tag{7}$$

where *MSE* is the mean squared error. The *BER* is computed as the percentage of error bits on the binarization of the watermark extraction w_i^*. In the testing, the *BER* is zero, indicating that the original and the extracted watermarks are identical. The testing *PSNR* is 39.72 dB, indicating a high fidelity of the marked-images, so that the hidden information cannot be noticed by human vision. A few testing examples with various image content and color are presented in Fig. 6, where we can observe high fidelity and full extraction. The watermark codes w_f^i do not directly reveal information about the watermark, which shows the perceivable randomness and decomposition learned by the proposed model.

C. The Proposed Scheme on Synthetic Images

To further validate that the image watermarking task is properly generalized, the proposed scheme is applied for synthetic RGB cover-images and watermarks. The results of

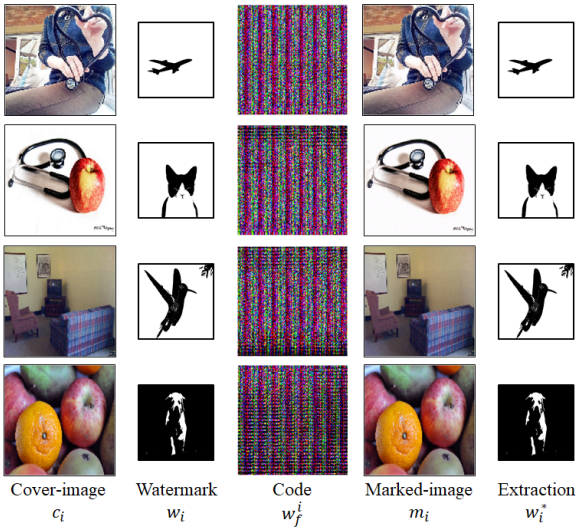

Fig. 6: A few testing examples of the proposed scheme with various image content and color.

the blank cover-images and the random bits are presented below.

Fig. 7 illustrates the scenario of embedding binary watermarks into synthetic blank cover-images of black, red, white, green, and blue colors, respectively. Although the blank cover-images are not included in the training, the proposed scheme provides promising results. Applying blank cover-images is known to be extremely difficult in conventional watermarking methods due to the lack of psycho-visual information. However, unlike traditional methods that assign some unnoticeable portions of visual components as the watermark, the proposed deep learning model learned to apply the correlation between the features of space W_f and of M to indicate the watermark.

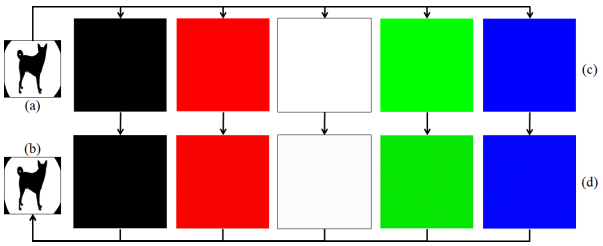

Fig. 7: Embedding watermarks into blank cover-image examples. (a) and (b): the embedded and extracted watermarks; (c) and (d): the five blank cover- and marked-images.

Fig. 8 shows an example of embedding a randomly generated binary image into a natural cover-image. To test the application scenarios where the watermarks are encrypted to random bits (besides the displayed example), 10,000 randomly generated bit sets are tested on 10,000 cover-images from the testing dataset. The average *BER* is 0.36%, which indicates that applying random binary bits as the watermark does not deteriorate the performance of our proposed solution.

D. The Robustness of the Proposed Scheme

The robustness of the proposed scheme against different distortions on the marked-image is evaluated by analyzing

Fig. 8: Embedding random bits as the watermark. (a) random bits; (b) a cover-image; (c) extraction; and (d) the marked-image.

the distortion tolerance range. Fig. 9 illustrates a few visual examples of the marked-images and their distortions.

Fig. 9: A few visual examples of the marked-images (top row) and their distortions (bottom row). The methods are implemented from left to right: Histogram Equalization, Gaussian Blur, Salt & Pepper Noise, and Cropping, respectively.

Due to the over-complete design and the invariance layer τ_{θ_5}, the proposed schemes can tolerate distortions at a very high percentage (see Fig. 10 (b) and (d) for an example of large cropping).

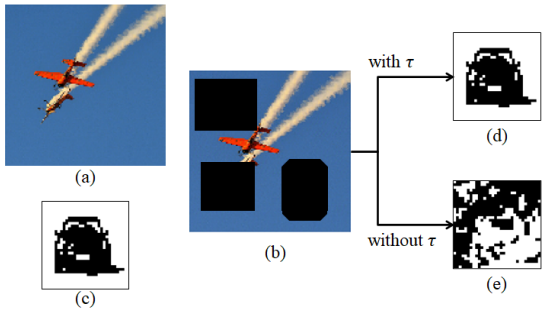

Fig. 10: An example of watermark extraction under large percentage cropping. (a) an example marked-image, (b) cropped (a), (c) original watermark, (d) extraction with τ, and (e) extraction without τ.

To demonstrate the importance of our core robustness provider, two control experiments extracting watermarks from distorted marked-images with and without τ_{θ_5} are conducted. Without τ_{θ_5}, the proposed scheme cannot extract the correct watermark (one example is shown in Fig.10), and the extraction from 10,000 testing attacked marked-images yields an average *BER* as high as 42.46%, which illustrates the significance of τ_{θ_5} if we compare to the results presented in Fig.11.

With τ_{θ_5}, distortions with swept-over parameters that control the attack strength are applied on the marked-images produced from the testing dataset. The watermark extraction *BER* caused by each distortion under each parameter is averaged over the testing dataset. The distortions with swept-over parameters versus the average *BER* are plotted in Fig. 11. Since focusing on image-processing attacks, the responses of the proposed scheme against some challenging image-processing attacks are discussed. The proposed scheme shows high tolerance range on these challenges, especially for cropping, salt-and-pepper noise, and JPEG compression. For example, the extracted watermarks have low average *BER*s as 7.8%, 11.6%, and 12.3% under severe distortions including a cropping discarding 65% of the marked image, a JPEG compression with a low quality factor 10, and a 90% salt-and-pepper noise. The attacks that randomly fluctuate the pixel values through image channels show a higher *BER*, including Gaussian additive noise and random noise that sets a random pixel to a random value. These extreme attacks can easily destroy most of the contents on the marked-image (see few examples in Fig. 12). Still, the proposed system achieves good performances when the marked-image contents are decently preserved, such as 14% *BER* on a 11% random noise.

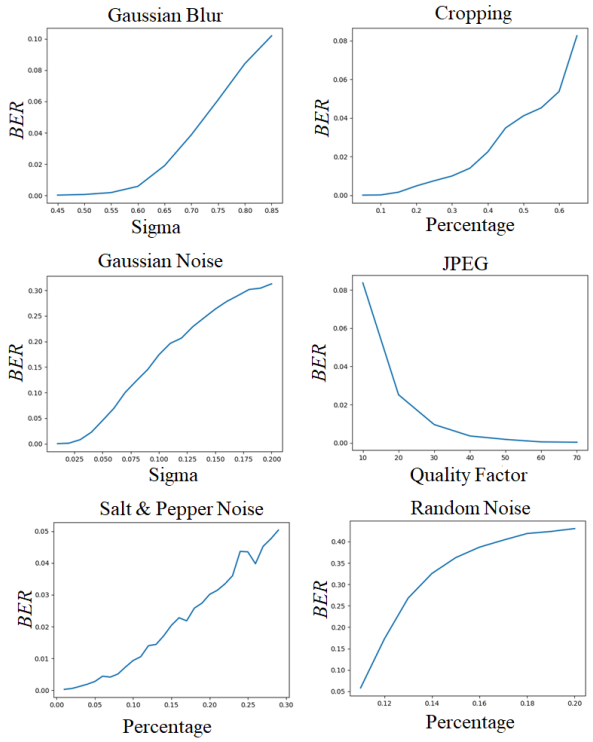

Fig. 11: Distortions with swept-over parameters versus average BER.

Fig. 12: An example of extreme distortions. (left): the marked-image; (middle): after Gaussian additive noise with variance 0.2; and (right): after 20% random noise.

As discussed in Table I, Mun's scheme [19] has the closest purpose with ours because it also achieves blindness and robustness simultaneously. Hence, we further compare our scheme with Mun's scheme for analysis. The comparison is performed on the same cover-image sets and the same watermark images as reported in Mun's scheme. To analyze the robustness of the proposed scheme, the extraction *BER*s under common image-processing attacks are shown in Table II.

The proposed scheme shows the advantages by covering more distortion categories in image-processing attacks and obtaining a lower *BER* under the same distortion parameters. Although Mun's method can tolerate geometric distortions while the proposed scheme cannot, Mun's method requires the presence of the distortions in the training phase for robustness. In the real world, there is no way to predict and enumerate all kinds of attacks.

E. A Case Study: Feasibility Test on Watermark Extraction from Camera Resamples

Currently, image watermarking applications are much different than they used to be when they were first developed. Early scenarios focus on copyright detection, while more recent real-world communication requirements introduce a challenging use-case: the watermark extraction from phone camera resamples of marked-images [28]. The challenges in this use-case arise because watermark extraction needs to handle multiple combinations of the distortions, such as optical tilt, quality degradation, compression, lens distortions, and lighting variation. Most existing approaches [29], [30], [31], [32] focus on the resamples of printed marked-images, not on phone resamples of a computer monitor screen. This use-case typically involves additional distortions, such as the Moiré pattern (*i.e.,* the RGB ripple), the refresh rate of the screen, and the spatial resolution of a monitor (some examples are mentioned in Fig. 13). We applied the proposed scheme as a major component in such scenarios since it is designed to reject all irrelevant noise instead of focusing on certain types of attacks. The outline of our scenario is shown in Fig. 13.

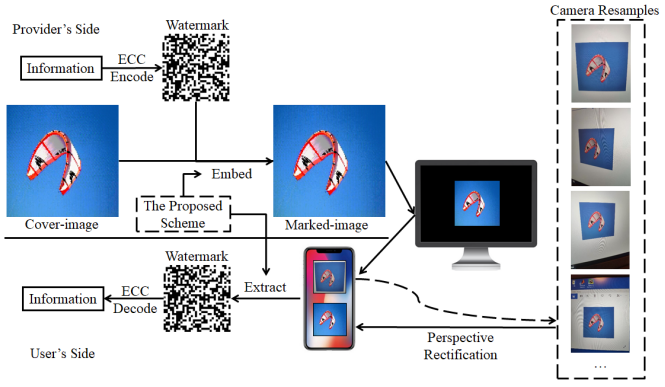

Fig. 13: The phone camera testing scenario.

An information provider prepares the information by encoding through an Error Correction Coding (ECC) technique. Although trained as an entire neural network, the proposed scheme is separated into the embedding components (μ_{θ_1} and σ_{θ_2}) and the extracting components (τ_{θ_5}, φ_{θ_3} and γ_{θ_4}). The marked-image can be obtained by embedding the encoded watermark into the cover-image using the trained embedding components. The marked-image that looks identical to the cover-image is distributed online and displayed on the user's screen. A user scans the marked-image with a phone to extract the hidden watermark through the extracting components.

The distortions occurred in this test can be divided into two categories: perspective and image-processing distortions. The major function of the proposed scheme in this scenario is to overcome the pixel-level modifications coming from image-processing distortions like compression, interpolation errors and the Moiré pattern. To concentrate the test on the proposed scheme, we simplified the solution of the perspective distortions, although this can be an entire challenging research track [33], [34].

With this setup, we develop a prototype for a user study. Fig. 14 (a) illustrates the Graphical User Interface (GUI) and a 32×16 sample information. Classic Reed Solomon (*RS*) code [35] is adopted as the ECC to protect the information. $RS(32, 16)$ is applied to protect each row of the 32×16 information so that the encoded information will be a 32×32 watermark satisfying the fixed watermarking capacity of the proposed scheme. In the watermark, each row is a codeword with data of length 16 and a parity of length 16, and hence can correct up to an error of length 8. Therefore, in this watermark of length 1,024 (32×32), up to 256 errors can be corrected if there are no more than 8 errors in each row [35]. Applying half of the bits as the parity, the watermarking payload is 512 bits. For the perspective distortion, four corners of the largest contour inside the Region Of Interest (ROI) are used to map the contoured content to a bird's-eye view (Fig. 14 (b)), and the watermark is extracted from the bird's-eye view rectification.

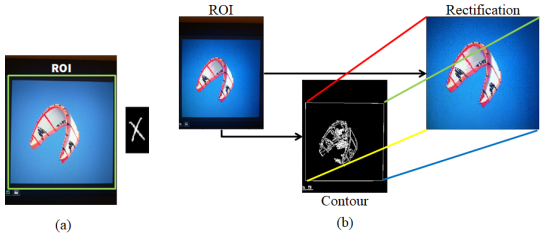

Fig. 14: The prototype. (a) The GUI and sample information; and (b) the simple rectification.

In this case study, five volunteers were invited to take a total of 25 photos of some marked-images displayed on a $2,560 \times 1,440$ screen using the camera on their mobile phones. The volunteers' phones were Google Pixel 3, Samsung Galaxy s9, iphone XR, Xiaomi 8, and iphone X. All the photos were taken under office light conditions. Volunteers were given two rules. First, the entire image should be placed as large as possible inside the ROI. As a prototype for demonstration, this rule facilitates our segmentation that the largest contour inside the ROI is the marked-image, so that this application can focus on the test of the proposed system instead of some complicated segmentation algorithms. In addition, placing the image largely in the ROI helps with the capture of desired details and features for the watermark extraction. Second, the camera should be kept as stable as possible. Although the proposed system tolerates some blurring effects, it is not designed to extract watermarks in high-speed motion.

Fig. 15 presents five watermark extractions, their *BERs*, and the corresponding ROIs. The closer up the photo is taken, the lower the error. Also, a lower error was observed with a greater parallel angle between the camera and the screen. The flashlight brings more errors due to over- or under-exposure to some image areas. Use of the flashlight in this application is optional because the screen has the back-lights. The average *BER* was 5.13% for the 25 images.

TABLE II: A quantitative comparison between the proposed scheme and Mun's scheme.

Method	BER (%) under the distortions					PSNR (dB)	Capacity (bits)
	HE	JPEG 10	Cropping 20%	S&P 5%	G. F. 10%		
Mun's	N/A	N/A	6.61	7.98	4.81	38.01	256
Ours	0.43	8.16	0	0.97	0	39.93	1,024

Note: N/A denotes "Not Applicable" that the robustness of an attack was not covered in [19]; HE denotes histogram equalization; JPEG 10 denotes a JPEG compression with quality factor 10; $S\&P$ denotes the salt-and-pepper noise; *G. F.* denotes Gaussian filtering.

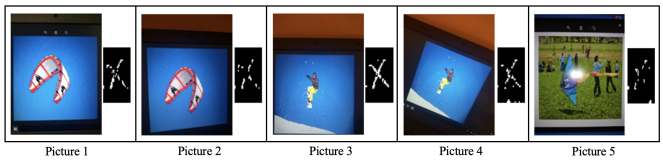

Fig. 15: Five examples of the watermark extractions before ECC and their ROIs. The left-hand side of each picture is the ROI, and the right-hand side is the extracted watermark. The *BER*s of the extractions from left to right are 3.71%, 4.98%, 1.07%, 4.30%, and 8.45%, respectively.

For a visual comparison, the displayed watermark extractions are the raw results before error correction. After executing $RS(32, 16)$, all the watermark extractions in the testing cases can be restored to the original information without errors, as shown in Fig. 14. The proposed scheme can successfully extract the watermark within one second because it only applies the trained weights on the marked-image rectification.

V. Applications of the Proposed Watermarking Scheme

In this section, we discuss different application scenarios, where the proposed image watermarking scheme can be applied to (i) authorized IoT device onboarding in Section V-A; (ii) creation of private communication channels in Section V-B; and (iii) authorized access to privileged content and services in Section V-C. We assume that a watermark is created based on credentials or secrets provided by users (e.g., passwords, cryptographic keys, or fingerprint scans). Our scenarios can also utilize the watermark extraction technique from camera resamples that we presented in Section IV-E.

A. Authorized IoT Device Onboarding

IoT devices need to be "onboarded" once are bought by users in terms of being associated with a home controller or some cloud service so that users can control them [36], [37].

Currently, the widely-used methods to onboard IoT devices are mostly out-of-band and include the use of QR codes physically printed on devices, pin codes, and serial numbers [38]. For example, once a user buys a smart IoT camera, he/she scans a QR code printed on the camera (or the packaging) with his/her mobile phone and through a mobile application, he/she connects the camera with a cloud service usually offered by its manufacturer. In this way, the user is able to watch the video stream capture from the camera.

Existing onboarding methods do not protect against unauthorized access. For example, attackers that have physical access to an IoT device can tamper it (e.g., install malware on the device) before the device is onboarded by its owner. The proposed image watermarking mechanism can be utilized to enable the onboarding of IoT devices only by the device owner. For example, user credentials can be embedded to a QR code, which will be physically printed on a device. Once a user receives his/her IoT device (*i.e.*, an IoT camera with a QR that has the user credential embedded), he/she takes a picture of the QR code with his/her mobile phone. To onboard the device, the user sends the taken QR picture along with his/her credentials to a server that runs the extraction via a deep neural network. The deep neural network verifies that the user indeed possesses the credentials (watermark) embedded into the QR code and authorizes the user to onboard the IoT device.

B. Creation of Private Communication Channels

The proposed image watermarking scheme can be used for the creation of private chatrooms and other communication channels. For instance, a chatroom organizer can collect the credentials of individuals that he/she would like to communicate with and create a QR code with this set of credentials embedded to the code. The created QR code can be uploaded on the Internet. Once an individual that has been included in the communication group scans the QR code with his/her mobile phone and provides his/her credential to the deep neural network, the network verifies that this individual indeed possesses credentials embedded into the QR code and authorizes the user to join the chatroom. Unauthorized Internet users (i.e., users that do not have their credentials embedded into the QR code watermark) might try to join the chatroom, but they will not be able to do so; even if such users take a photo of the QR code, they do not possess credentials that are embedded into this code; thus, the deep neural network will reject their requests to join the chatroom. The QR code with the embedded watermark will be publicly available and can be accessed by all Internet users, however, only the authorized users will be able to join the chatroom and communicate with each other.

C. Authorized Access to Privileged Content and Services

The proposed image watermarking scheme can be also utilized for a broader scope of applications, where access to privileged content and/or services is desirable. The content producer or service provider will create cover-images that have the credentials of authorized users embedded. As a result, image watermarking can be used as an access control mechanism, where access to certain pieces of (privileged) content and/or services are restricted only to authorized users; *i.e.*, users that have their credentials (watermark) embedded into the marked image. Similarly, when authorized users send the marked image and the embedded credentials to the deep neural network, the network will be able to extract the watermark

only if the user possesses the proper credentials. Only users that possess the credentials (watermark) embedded into the marked image will be allowed to access the privileged content.

VI. Conclusion

This paper introduces an automated and robust image watermarking scheme using deep convolutional neural networks. The proposed blind image watermarking scheme exploits the fitting ability of deep neural networks to generalize image watermarking algorithms, shows an architecture that trains in an unsupervised manner for watermarking tasks, and achieves its robustness property without requiring prior knowledge of possible distortions on the marked-image. Experimentally, we have not only reported the promising performances for individual common attacks, but also have demonstrated that the proposed scheme has the ability and the potential to help combinative, cutting-edge, and challenging camera applications, which has confirmed the superiority of the proposed scheme. Our future work includes tackling geometric and perspective distortions by the deep neural networks inside the scheme, and refining the scheme architecture, objective and loss function by different methods like ablation studies.

References

[1] H. Berghel and L. O'Gorman, "Protecting ownership rights through digital watermarking," *Computer*, vol. 29, no. 7, pp. 101–103, 1996.

[2] I. J. Cox, M. L. Miller, and A. L. McKellips, "Watermarking as communications with side information," *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1127–1141, 1999.

[3] F. Y. Shih, *Digital watermarking and steganography: fundamentals and techniques*. CRC press, 2017.

[4] I. J. Cox, J. Kilian, F. T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," *IEEE transactions on image processing*, vol. 6, no. 12, pp. 1673–1687, 1997.

[5] I. Cox, M. Miller, J. Bloom, J. Fridrich, and T. Kalker, *Digital watermarking and steganography*. Morgan kaufmann, 2007.

[6] X. Kang, J. Huang, Y. Q. Shi, and Y. Lin, "A dwt-dft composite watermarking scheme robust to both affine transform and jpeg compression," *IEEE transactions on circuits and systems for video technology*, vol. 13, no. 8, pp. 776–786, 2003.

[7] S. Craver, N. Memon, B.-L. Yeo, and M. M. Yeung, "Resolving rightful ownerships with invisible watermarking techniques: Limitations, attacks, and implications," *IEEE Journal on selected Areas in communications*, vol. 16, no. 4, pp. 573–586, 1998.

[8] H. Kandi, D. Mishra, and S. R. S. Gorthi, "Exploring the learning capabilities of convolutional neural networks for robust image watermarking," *Computers & Security*, vol. 65, pp. 247–268, 2017.

[9] V. Vukotić, V. Chappelier, and T. Furon, "Are deep neural networks good for blind image watermarking?" in *2018 IEEE International Workshop on Information Forensics and Security (WIFS)*. IEEE, 2018, pp. 1–7.

[10] D. Li, L. Deng, B. B. Gupta, H. Wang, and C. Choi, "A novel cnn based security guaranteed image watermarking generation scenario for smart city applications," *Information Sciences*, vol. 479, pp. 432–447, 2019.

[11] A. Fierro-Radilla, M. Nakano-Miyatake, M. Cedillo-Hernandez, L. Cleofas-Sanchez, and H. Perez-Meana, "A robust image zero-watermarking using convolutional neural networks," in *2019 7th International Workshop on Biometrics and Forensics (IWBF)*. IEEE, 2019, pp. 1–5.

[12] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 2016, pp. 372–387.

[13] Y. Huang, B. Niu, H. Guan, and S. Zhang, "Enhancing image watermarking with adaptive embedding parameter and psnr guarantee," *IEEE Transactions on Multimedia*, vol. 21, no. 10, pp. 2447–2460, 2019.

[14] Z. Su, G. Zhang, F. Yue, L. Chang, J. Jiang, and X. Yao, "Snr-constrained heuristics for optimizing the scaling parameter of robust audio watermarking," *IEEE Transactions on Multimedia*, vol. 20, no. 10, pp. 2631–2644, 2018.

[15] Z. Chen, L. Li, H. Peng, Y. Liu, and Y. Yang, "A novel digital watermarking based on general non-negative matrix factorization," *IEEE Transactions on Multimedia*, vol. 20, no. 8, pp. 1973–1986, 2018.

[16] Z. Wang, N. Gao, X. Wang, X. Qu, and L. Li, "Sstegan: Self-learning steganography based on generative adversarial networks," in *International Conference on Neural Information Processing*. Springer, 2018, pp. 253–264.

[17] S. Baluja, "Hiding images in plain sight: Deep steganography," in *Advances in Neural Information Processing Systems*, 2017, pp. 2069–2079.

[18] W.-H. Kim, J.-U. Hou, S.-M. Mun, and H.-K. Lee, "Convolutional neural network architecture for recovering watermark synchronization," *arXiv preprint arXiv:1805.06199*, 2018.

[19] S.-M. Mun, S.-H. Nam, H. Jang, D. Kim, and H.-K. Lee, "Finding robust domain from attacks: A learning framework for blind watermarking," *Neurocomputing*, vol. 337, pp. 191–202, 2019.

[20] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science*, vol. 313, no. 5786, pp. 504–507, 2006.

[21] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[22] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio, "Contractive auto-encoders: Explicit invariance during feature extraction," in *Proceedings of the 28th International Conference on International Conference on Machine Learning*. Omnipress, 2011, pp. 833–840.

[23] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient backprop," in *Neural networks: Tricks of the trade*. Springer, 2012, pp. 9–48.

[24] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

[25] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," Citeseer, Tech. Rep., 2009.

[26] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[27] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[28] "Digimarc: A digital watermarking foundation," https://www.digimarc.com/about/technology/about-digital-watermarking, 2019.

[29] A. Pramila, A. Keskinarkaus, and T. Seppänen, "Increasing the capturing angle in print-cam robust watermarking," *Journal of Systems and Software*, vol. 135, pp. 205–215, 2018.

[30] W.-g. Kim, S. H. Lee, and Y.-s. Seo, "Image fingerprinting scheme for print-and-capture model," in *Pacific-Rim Conference on Multimedia*. Springer, 2006, pp. 106–113.

[31] T. Yamada and M. Kamitani, "A method for detecting watermarks in print using smart phone: finding no mark," in *Proceedings of the 5th Workshop on Mobile Video*. ACM, 2013, pp. 49–54.

[32] L. A. Delgado-Guillen, J. J. Garcia-Hernandez, and C. Torres-Huitzil, "Digital watermarking of color images utilizing mobile platforms," in *2013 IEEE 56th International Midwest Symposium on Circuits and Systems (MWSCAS)*. IEEE, 2013, pp. 1363–1366.

[33] M. Zhao, Y. Wu, S. Pan, F. Zhou, B. An, and A. Kaup, "Automatic registration of images with inconsistent content through line-support region segmentation and geometrical outlier removal," *IEEE Transactions on Image Processing*, vol. 27, no. 6, pp. 2731–2746, 2018.

[34] M. Jaderberg, K. Simonyan, and A. Zisserman, "Spatial transformer networks," in *Advances in neural information processing systems*, 2015, pp. 2017–2025.

[35] I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields," *Journal of the society for industrial and applied mathematics*, vol. 8, no. 2, pp. 300–304, 1960.

[36] S. Mastorakis, A. Mtibaa, J. Lee, and S. Misra, "ICedge: When Edge Computing Meets Information-Centric Networking," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4203–4217, 2020.

[37] S. Mastorakis and A. Mtibaa, "Towards service discovery and invocation in data-centric edge networks," in *2019 IEEE 27th International Conference on Network Protocols (ICNP)*. IEEE, 2019, pp. 1–6.

[38] S. Latvala, M. Sethi, and T. Aura, "Evaluation of out-of-band channels for iot security," *SN Computer Science*, vol. 1, no. 1, p. 18, 2020.

Xin Zhong received his Ph.D. from New Jersey Institute of Technology, Newark, New Jersey, USA, in 2018. He is currently an assistant professor with the Robotics, Networking, Artificial intelligence (R. N. A.) Laboratory, in the department of Computer Science, University of Nebraska Omaha. His research interests include image watermarking, computer vision, image processing and analysis, pattern recognition, machine learning, and deep learning.

Pei-Chi Huang received her Ph.D. from the University of Texas at Austin, Texas, USA, in 2017. She is currently an assistant professor with the Robotics, Networking, Artificial intelligence (R. N. A.) Laboratory, in the department of Computer Science, University of Nebraska Omaha. Her research interests include cyber-physical systems, machine learning, and robotics.

Spyridon Mastorakis received his Ph.D. from the University of California, Los Angeles (UCLA), Los Angeles, California, USA, in 2019. He is currently an assistant professor with the Robotics, Networking, Artificial intelligence (R. N. A.) Laboratory, in the department of Computer Science, the University of Nebraska Omaha. His research interests include network systems and protocols, Internet architectures (such as Information-Centric Networking and Named-Data Networking), and edge computing.

Frank Y. Shih received Ph.D. from Purdue University, West Lafayette, Indiana, U.S.A., in 1987. He is currently a professor with Artificial Intelligent and Computer Vision Lab, in the Department of Computer Science, New Jersey Institute of Technology, Newark, New Jersey. He has authored six books including "Digital Watermarking and Steganography," "Image Processing and Mathematical Morphology," "Image Processing and Pattern Recognition," and "Multimedia Security: Watermarking, Steganography, and Forensics." His research interests include artificial intelligence, deep learning, image processing, watermarking and steganography, digital forensics, pattern recognition, bioinformatics, biomedical engineering, fuzzy logic, and neural networks.